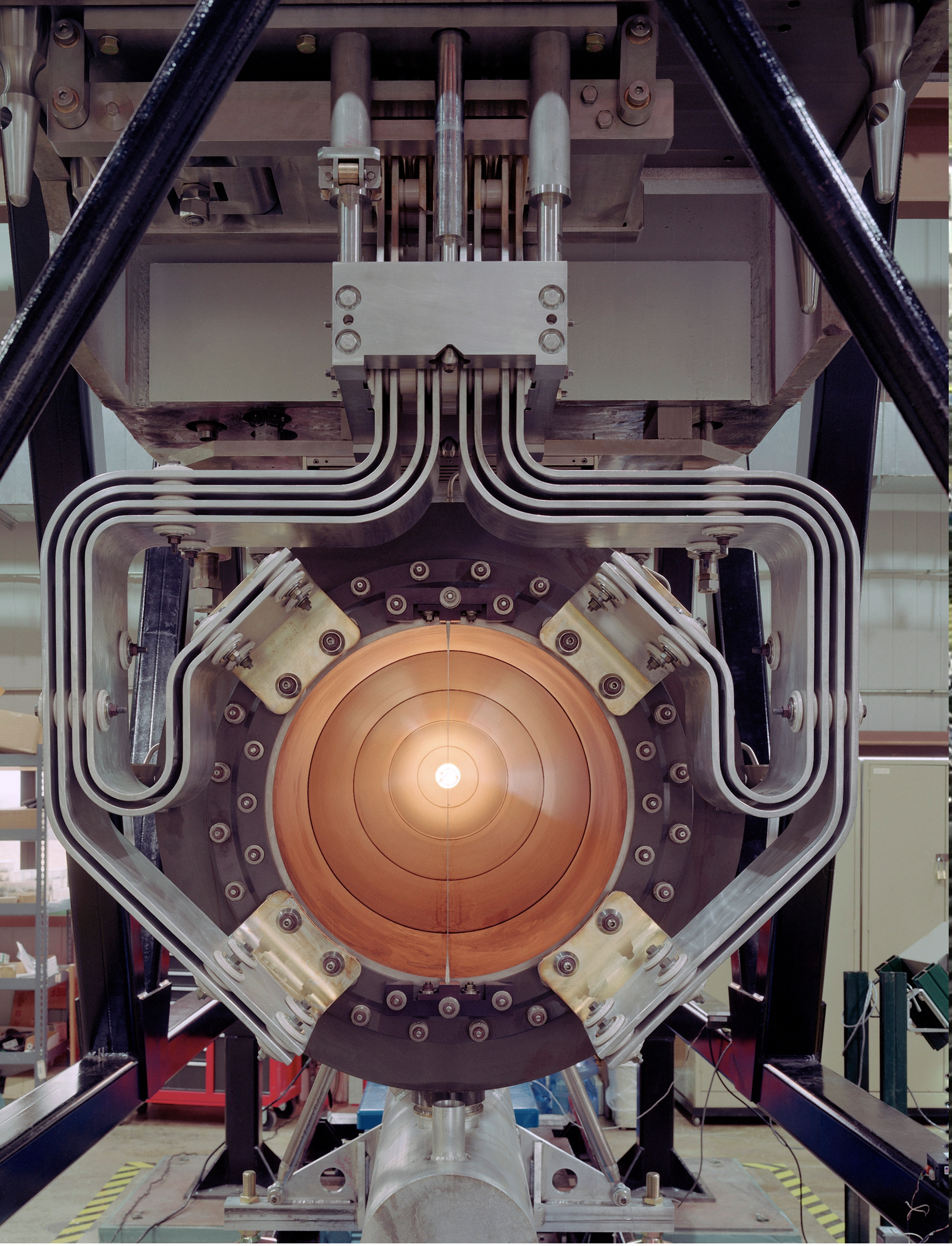
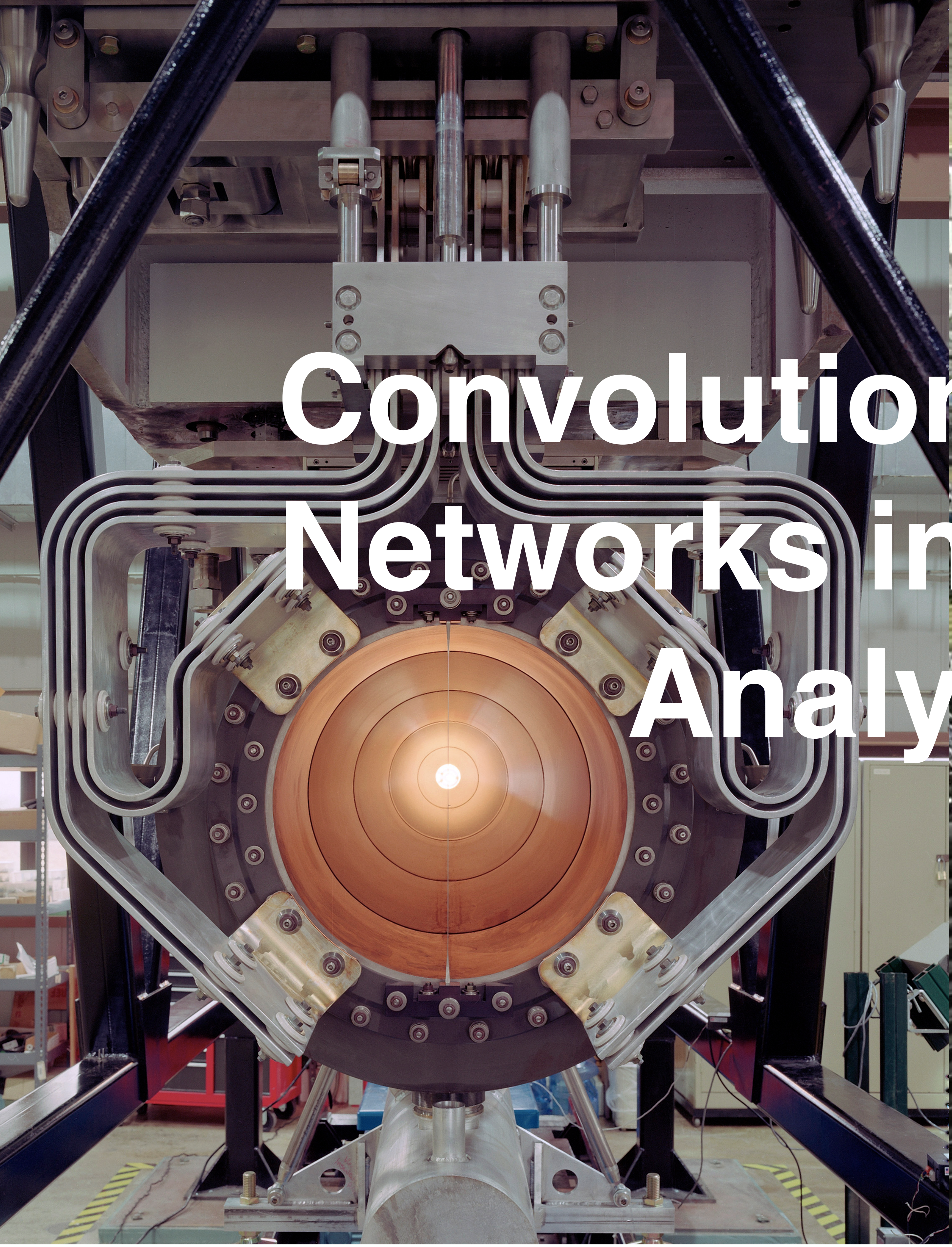


WILLIAM & MARY
CHARTERED 1693

Alexander Radovic
College of William and Mary

Convolutional Neural Networks in Neutrino Analyses



Alexander Radovic
College of William and Mary

“I have done a terrible thing, I have postulated a particle that cannot be detected.”

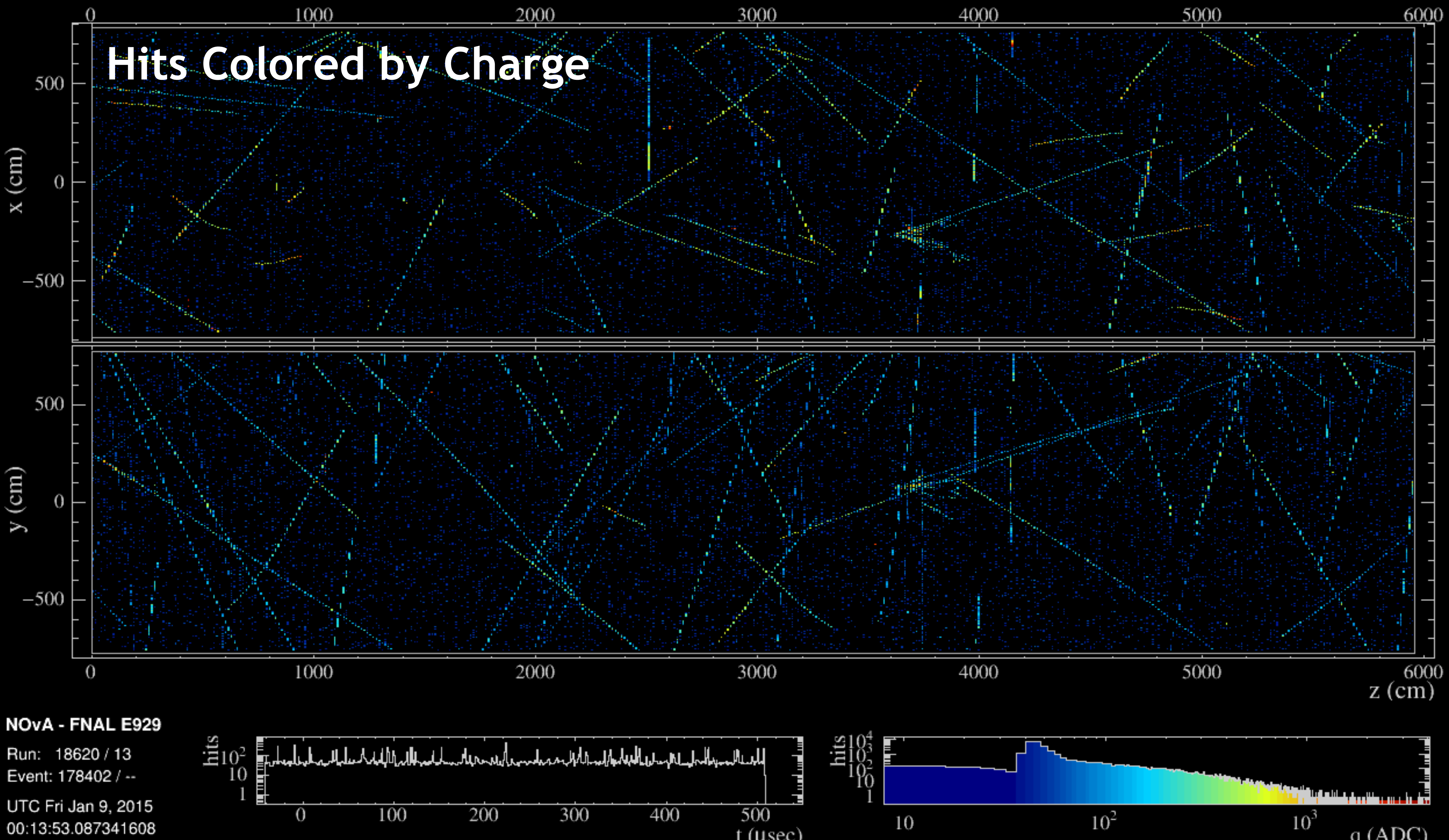
-Wolfgang Pauli

"I have done a terrible thing, I have postulated a particle that cannot be detected."

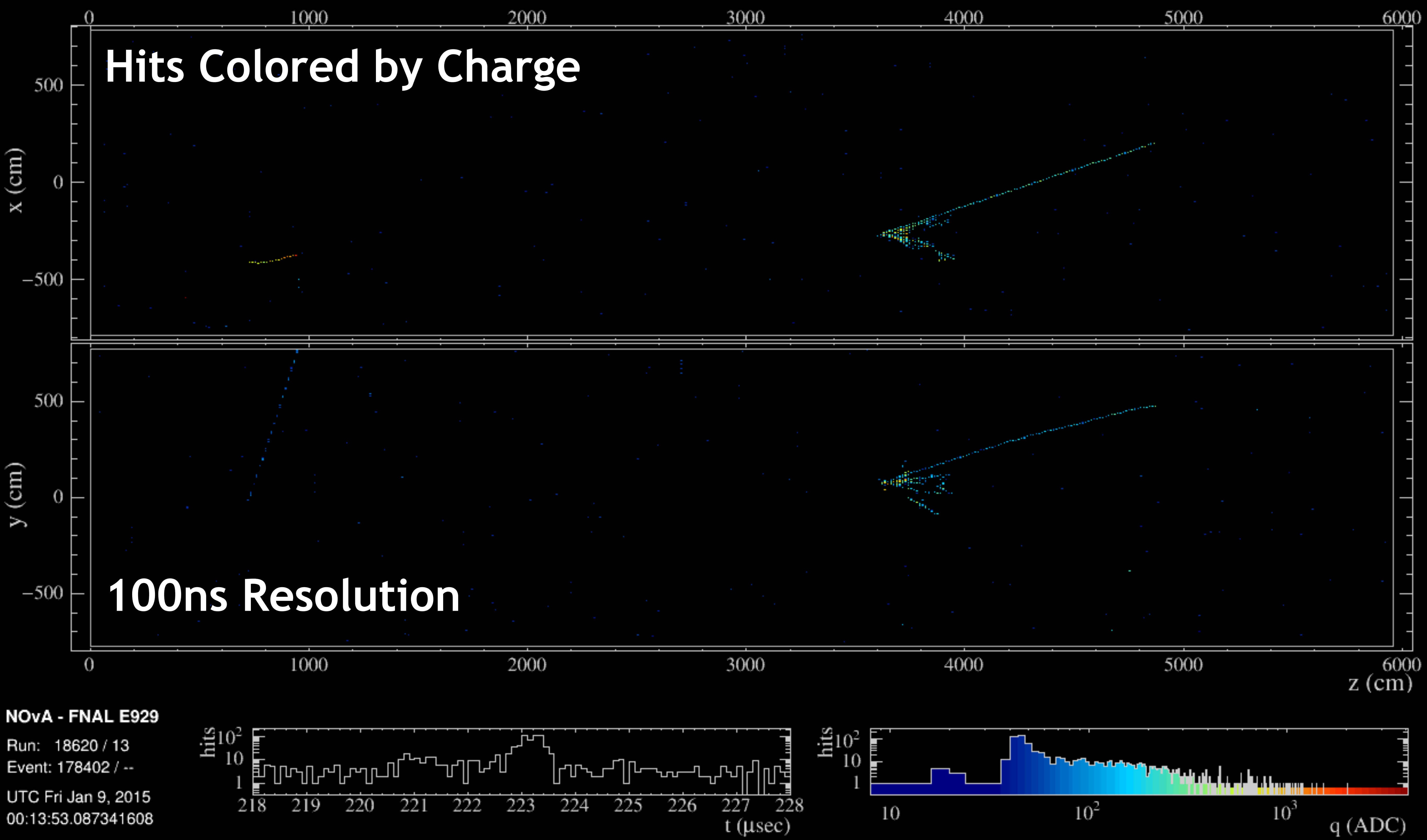
can

-Wolfgang Pauli

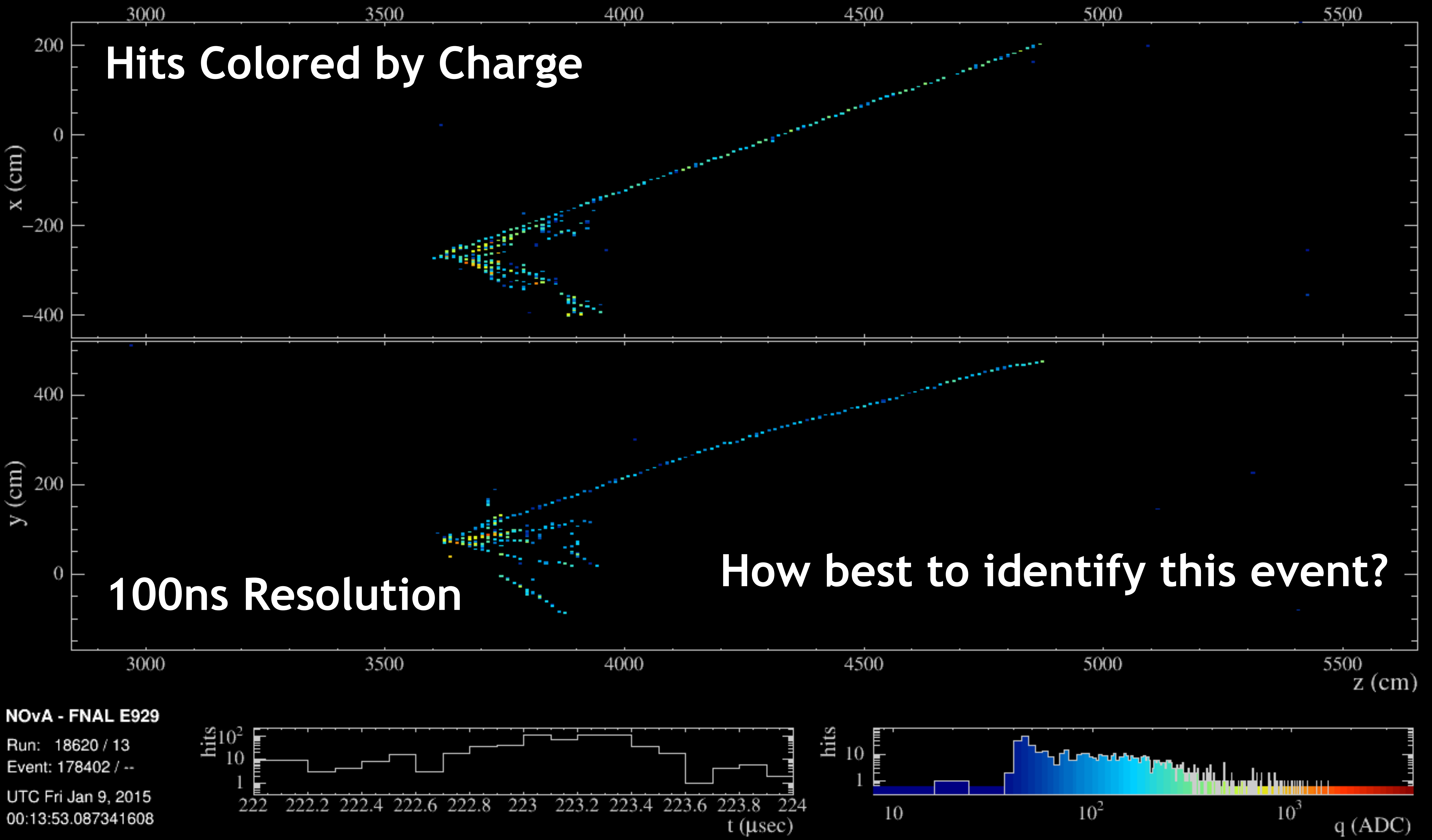
550 μs exposure of the NOvA Far Detector



Time-zoom on 10 μs interval during NuMI beam pulse

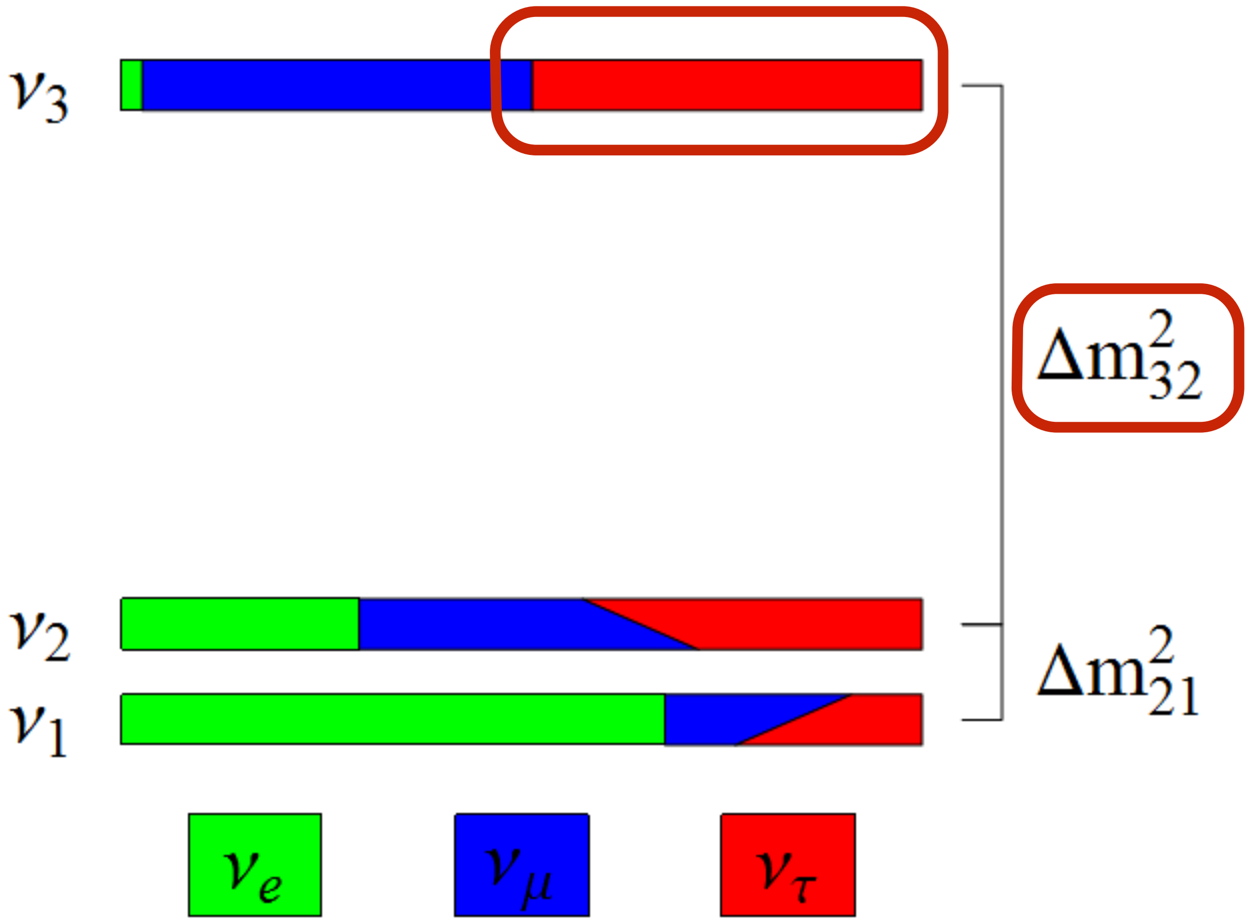


Close-up of neutrino interaction in the NOvA Far Detector



NOvA

NOvA Physics Goals



Precise measurements:

Δm^2_{32} and $\sin^2(2\theta_{23})$ for neutrinos and antineutrinos

Strong Constraints on:

θ_{23} octant

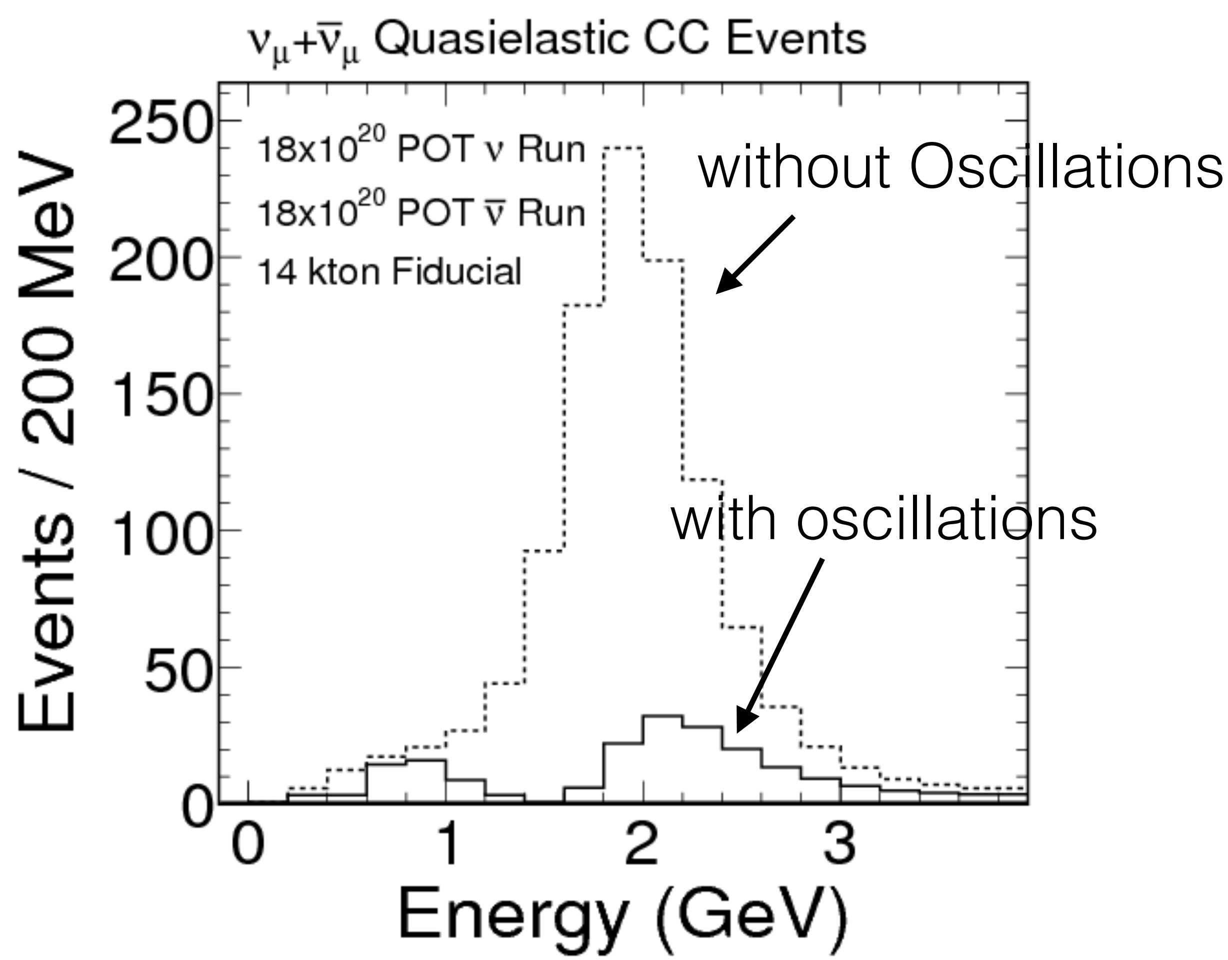
δ_{cp}

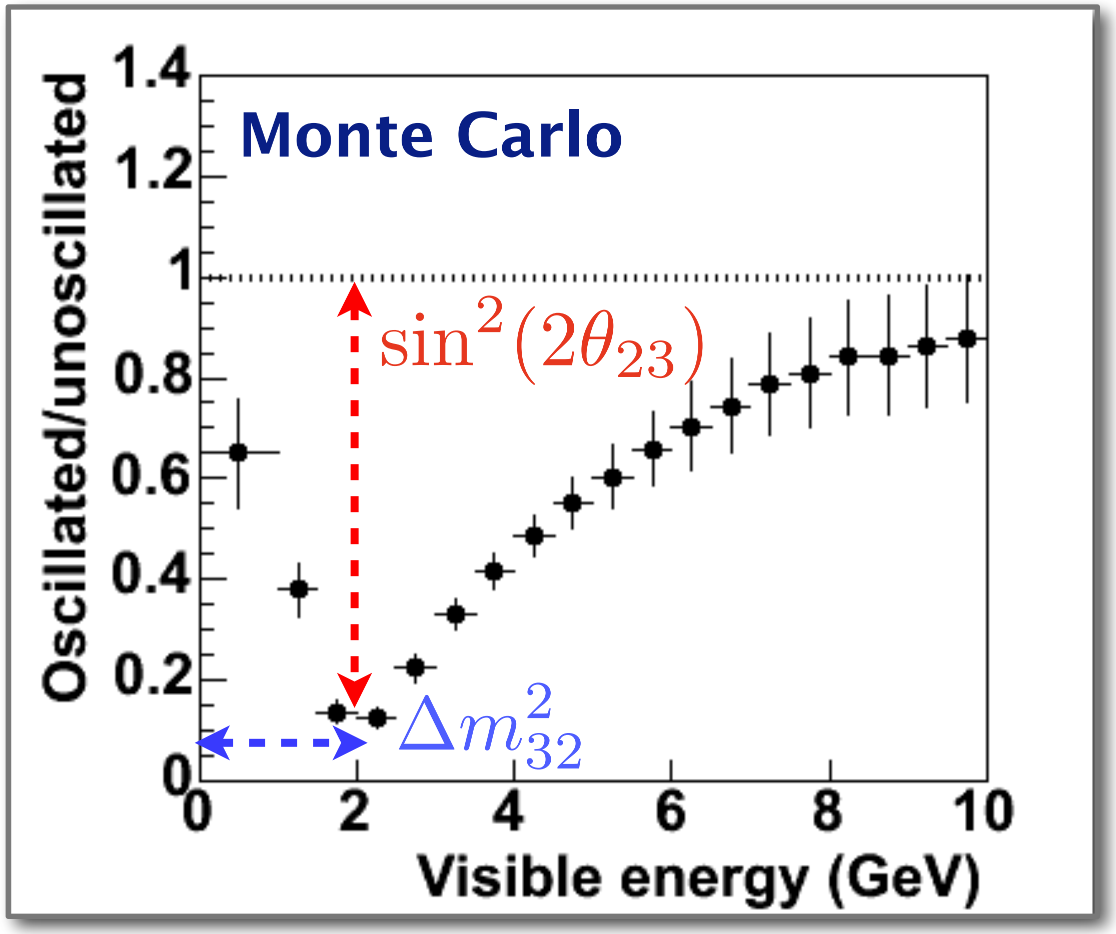
mass hierarchy

ν_μ Disappearance

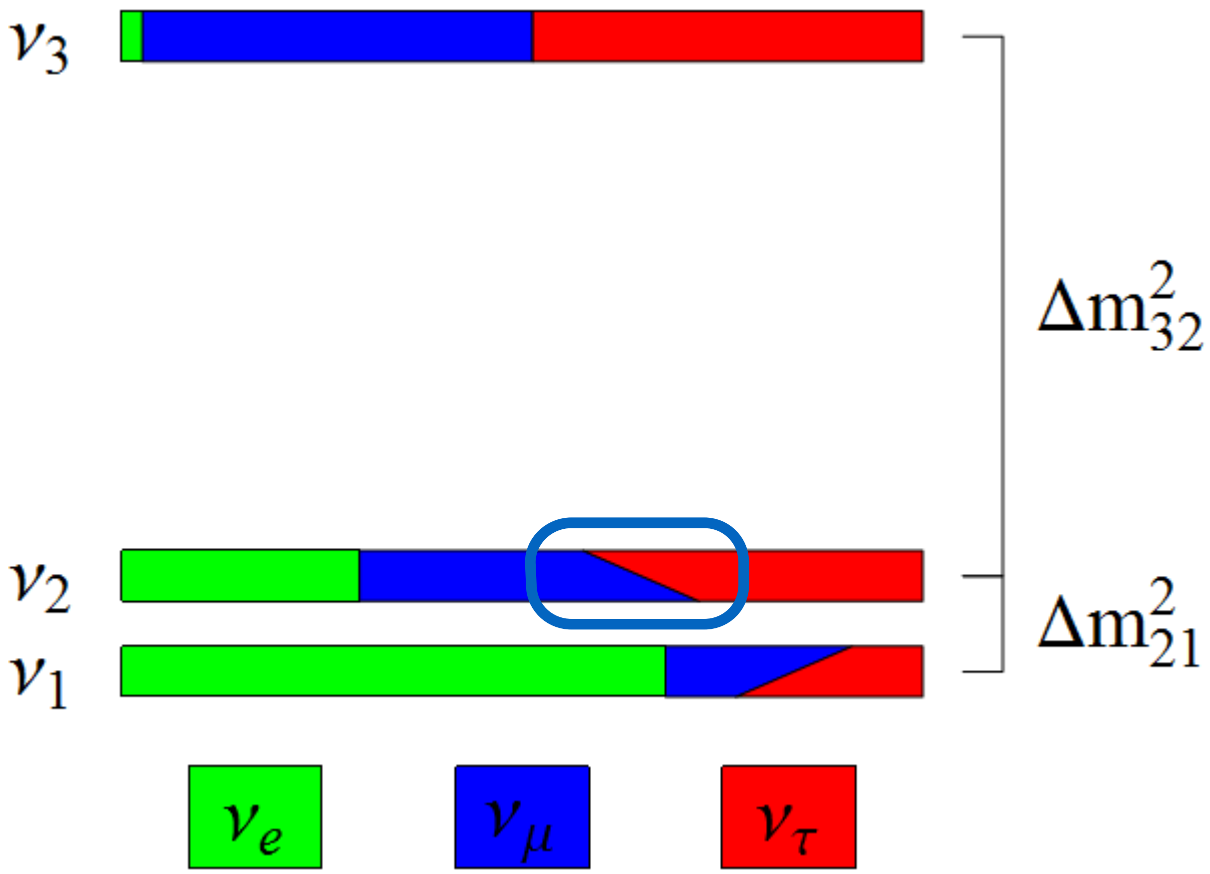
- Far detector prediction from near detector is compared to far detector measurement
- Neutrino oscillations deplete rate and distort the energy spectrum

$$P(\nu_\mu \to \nu_\mu) \approx 1 - \sin^2(2\theta_{23}) \sin^2\left(\frac{1.27\Delta m^2_{atm} L}{E}\right)$$





NOvA Physics Goals



Precise measurements:

Δm^2_{32} and $\sin^2(2\theta_{23})$ for neutrinos and antineutrinos

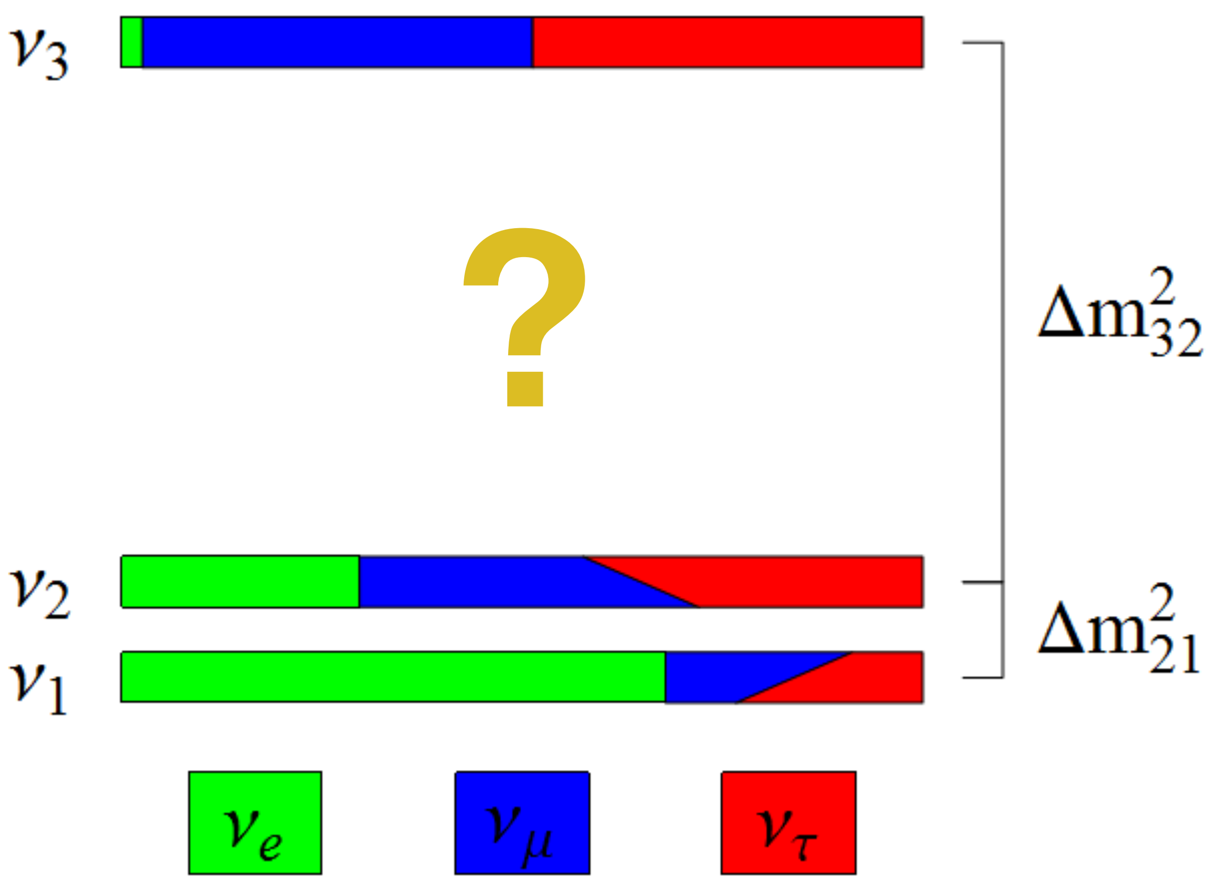
Strong Constraints on:

θ_{23} octant

δ_{cp}

mass hierarchy

NOvA Physics Goals



Precise measurements:

Δm^2_{32} and $\sin^2(2\theta_{23})$ for neutrinos and antineutrinos

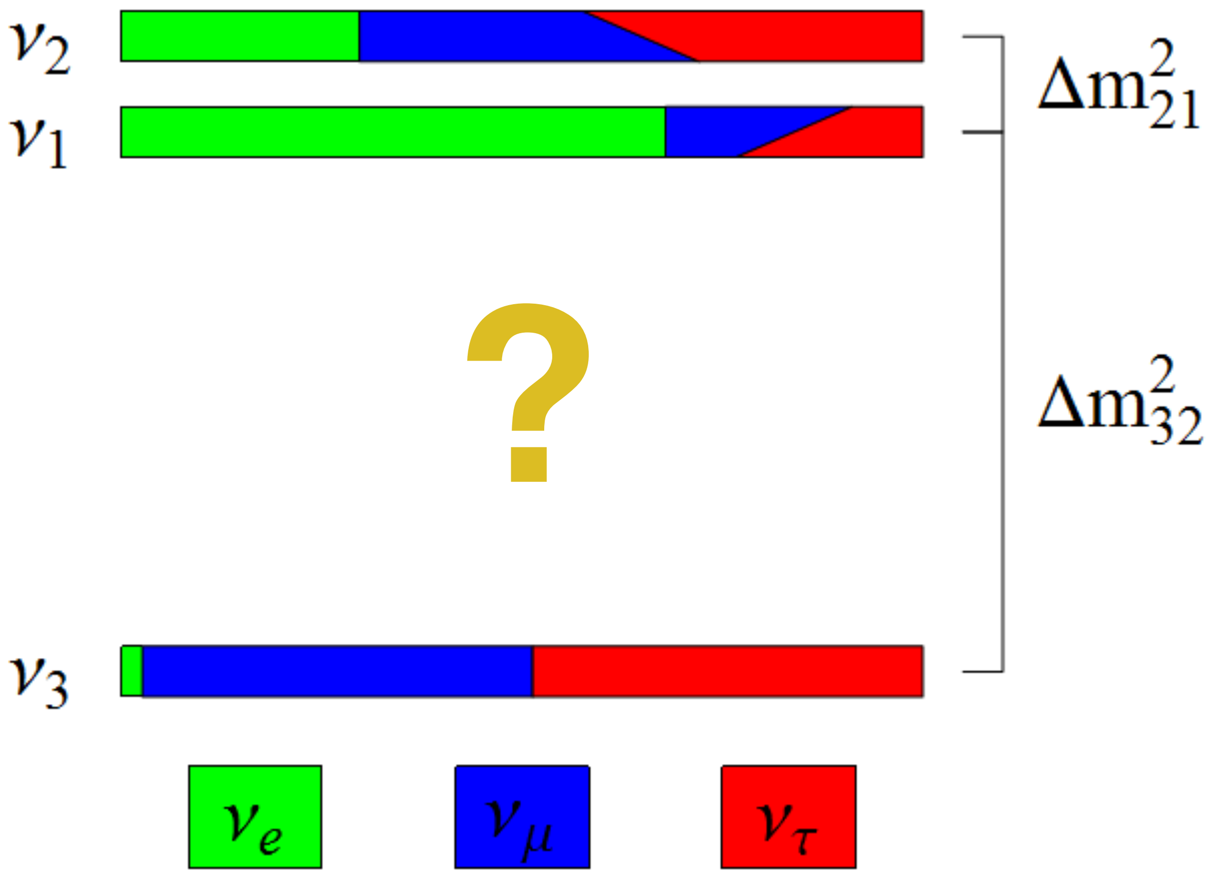
Strong Constraints on:

θ_{23} octant

δ_{cp}

mass hierarchy

NOvA Physics Goals



Precise measurements:

Δm^2_{32} and $\sin^2(2\theta_{23})$ for neutrinos and antineutrinos

Strong Constraints on:

θ_{23} octant

δ_{cp}

mass hierarchy

ν_e Appearance

By measuring beam muon neutrinos which have oscillated to electron neutrinos we gain the power to constrain:

θ_{23} octant

δ_{cp}

mass hierarchy

$$P(\nu_\mu \rightarrow \nu_e) \approx \left| \sqrt{P_{atm}} e^{-i\left(\frac{\Delta m^2_{32} L}{4E} + \delta_{cp}\right)} + \sqrt{P_{sol}} \right|^2$$

$$P_{atm} = \sin^2\theta_{23} \sin^2 2\theta_{13} \sin^2 \frac{\Delta m^2_{31} L}{4E}$$

Solar term contributes <1% at ~400 L/E

ν_e Appearance

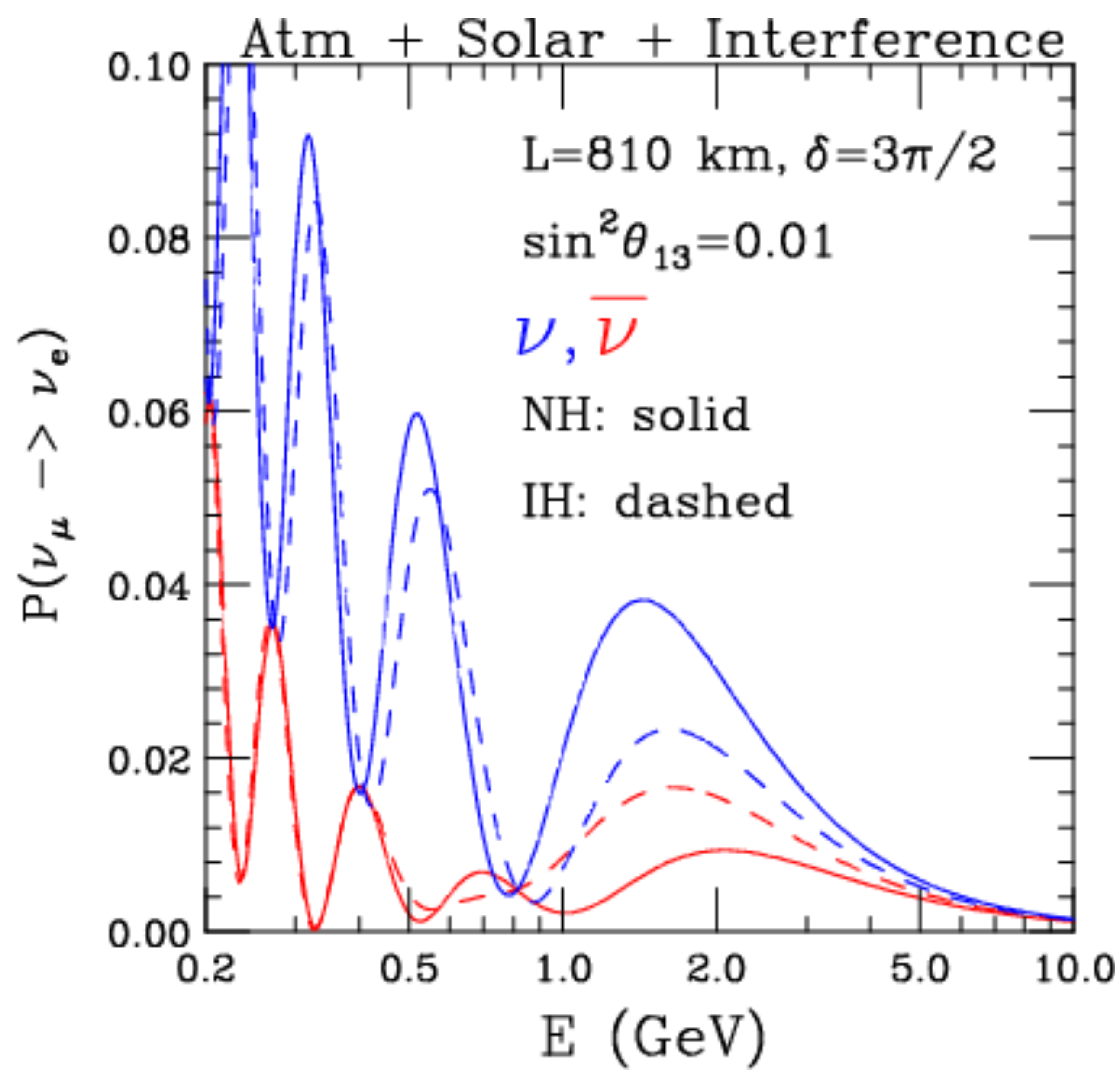
By measuring beam muon neutrinos which have oscillated to electron neutrinos we gain the power to constrain:

θ_{23} octant

δ_{cp}

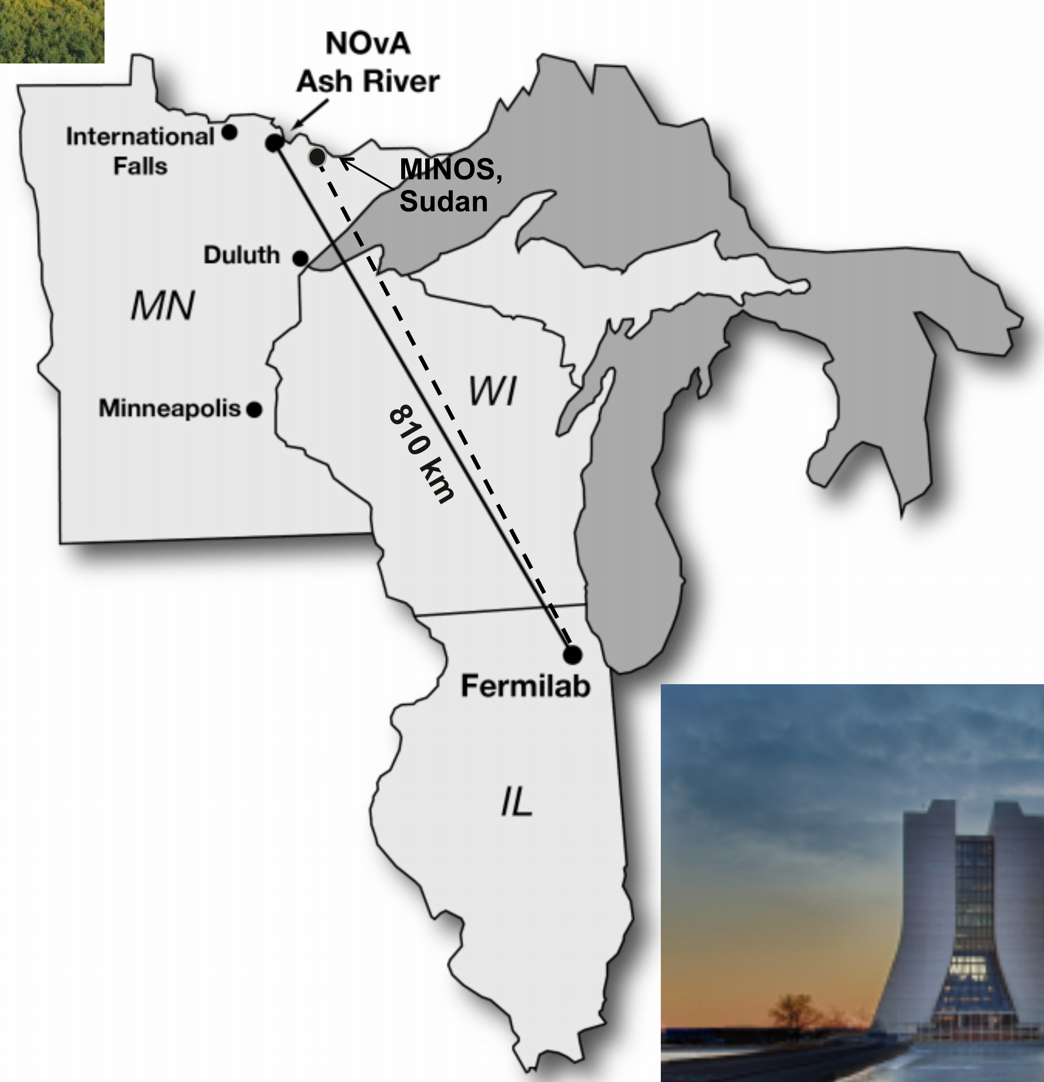
mass hierarchy

Electron neutrinos experience an extra interaction as they pass through matter, modifying oscillation probabilities, giving us a window into the mass hierarchy

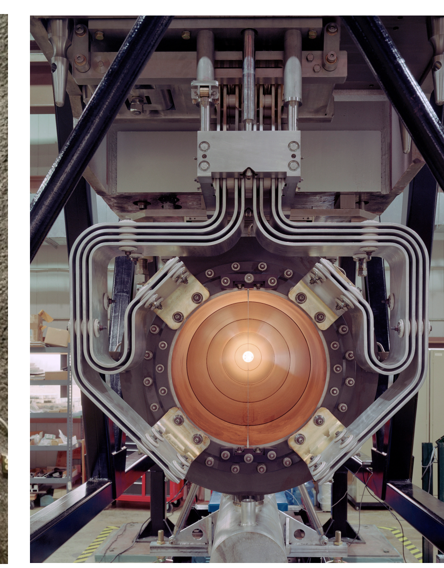


From S. Parke, "Neutrino Oscillation Phenomenology" in Neutrino Oscillations: Present Status and Future Plans

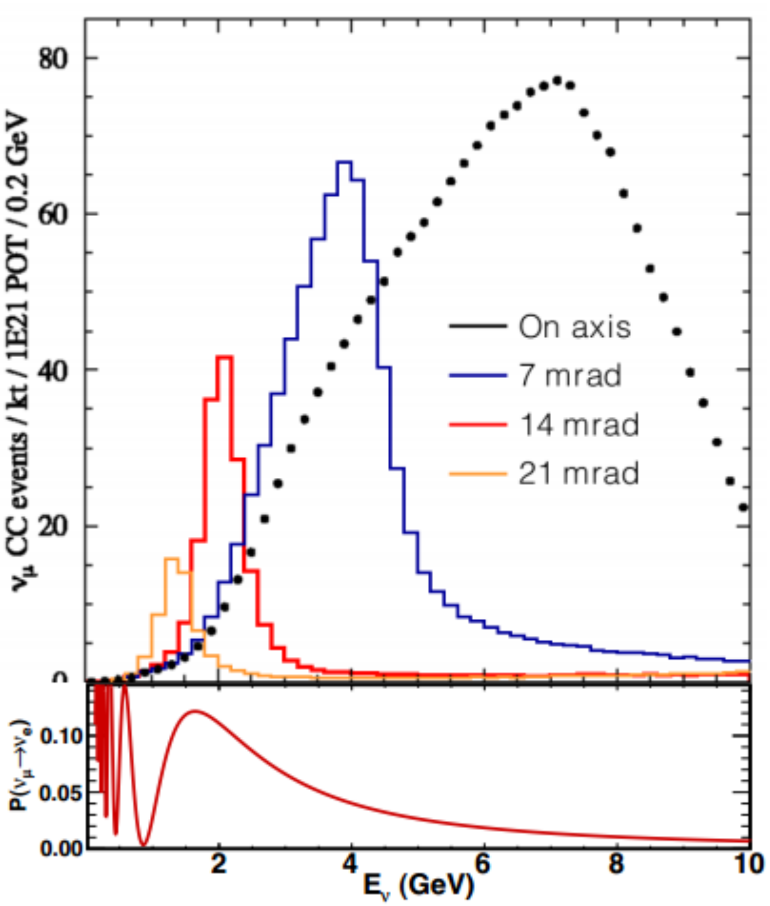
The NOvA Experiment

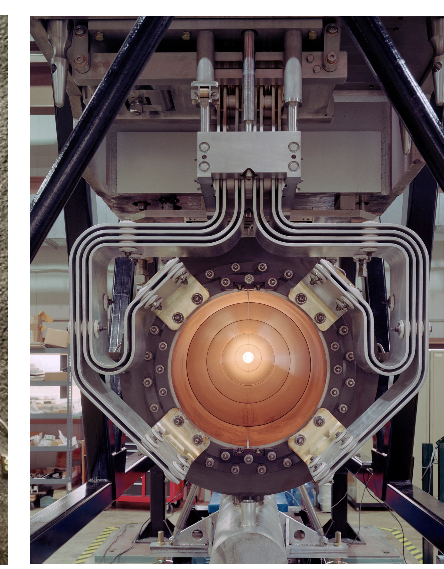


The NOvA Experiment

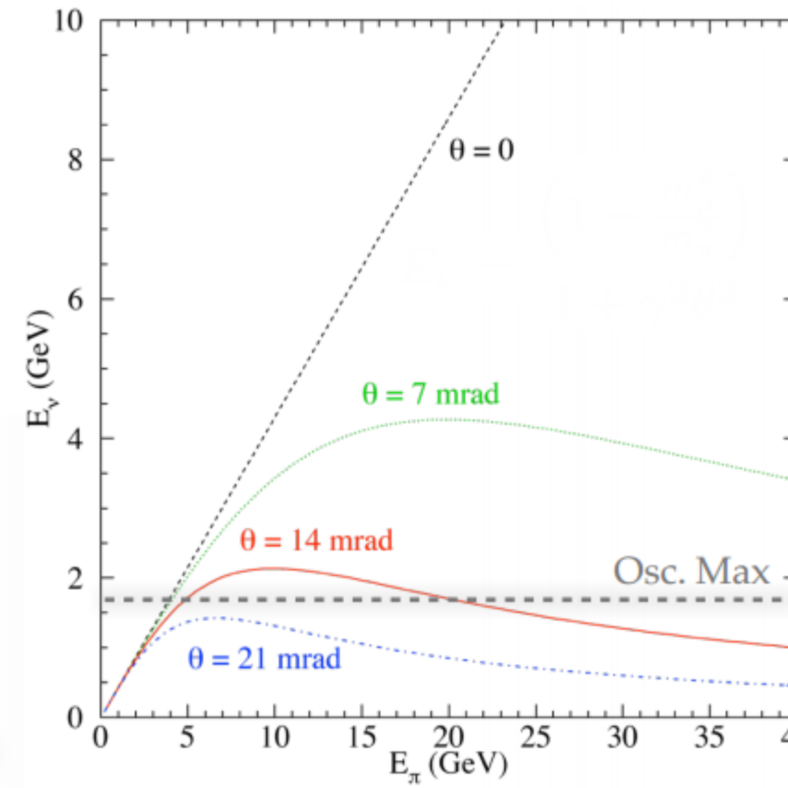


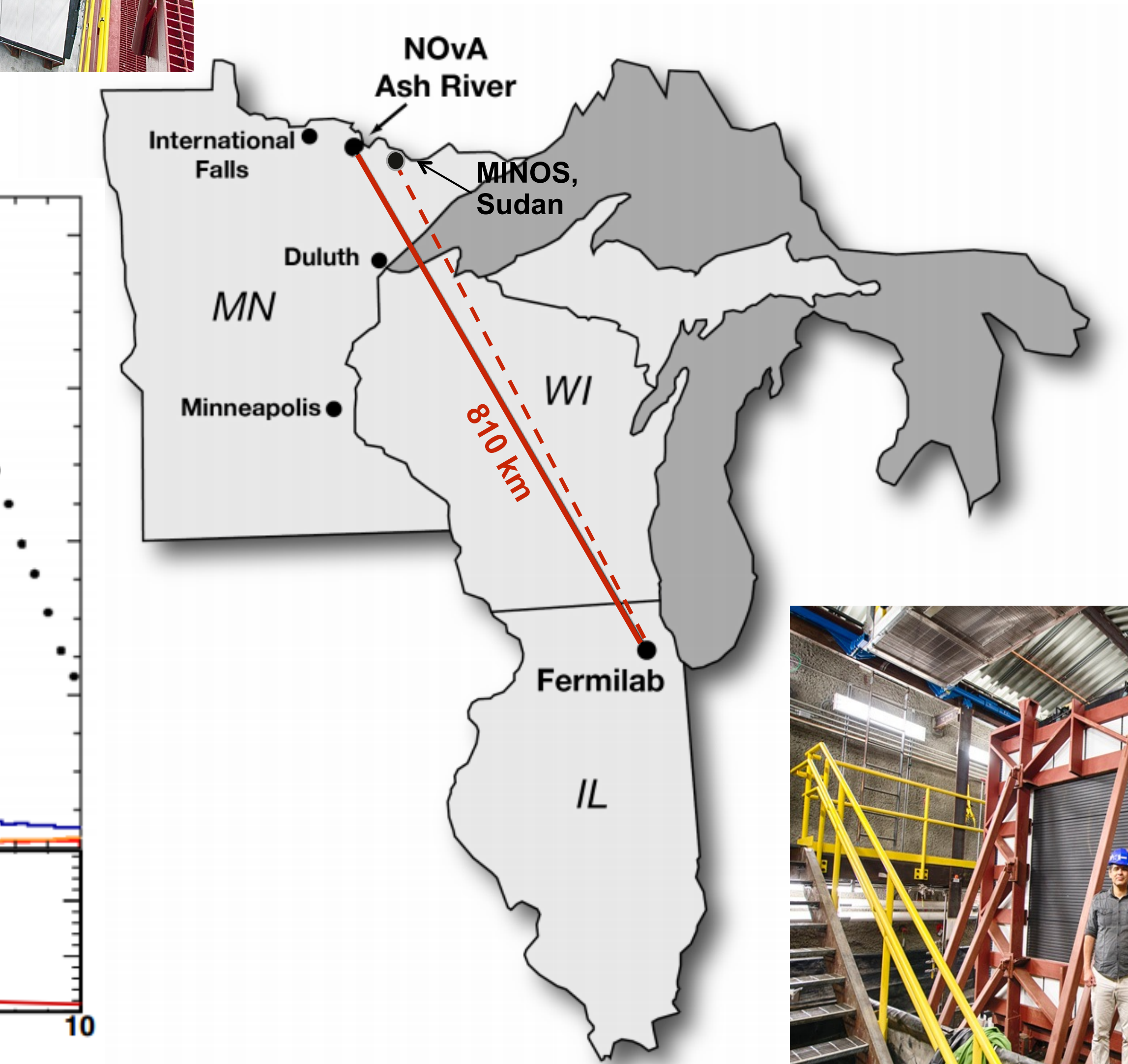
The NOvA Experiment

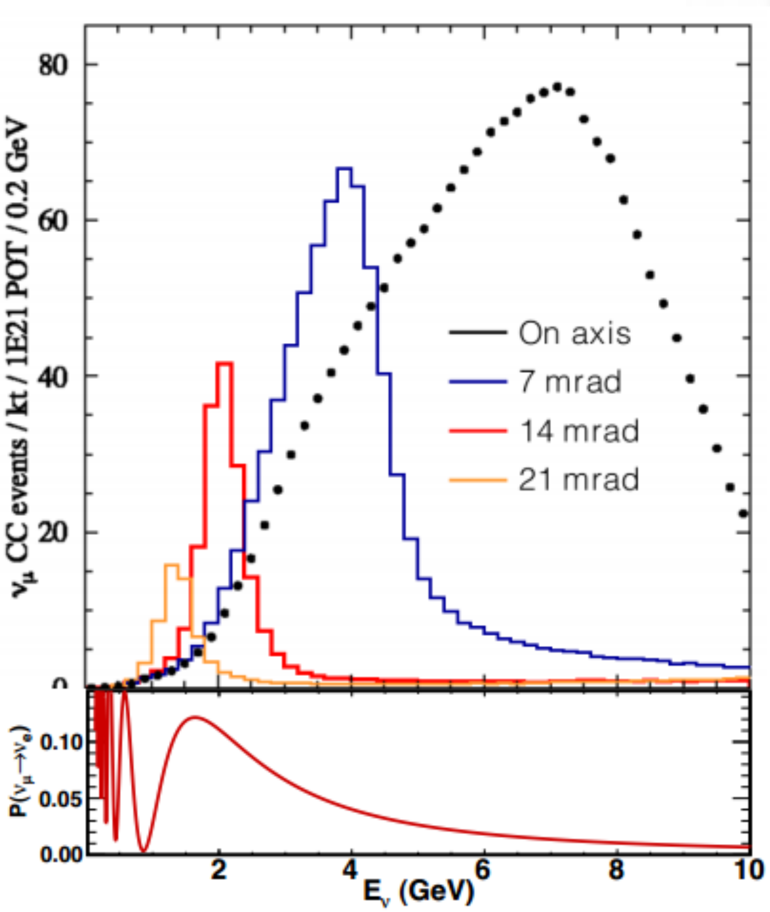


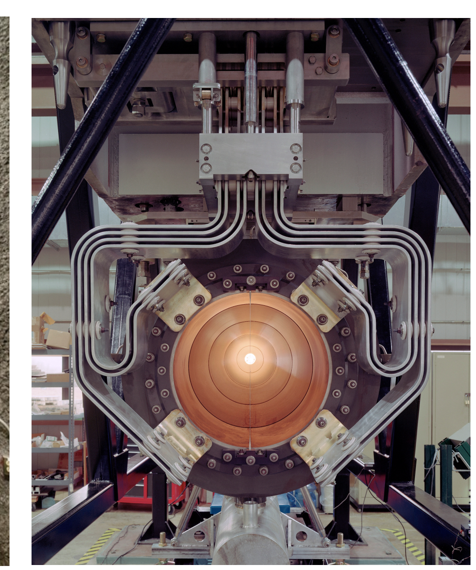


The NOvA Experiment

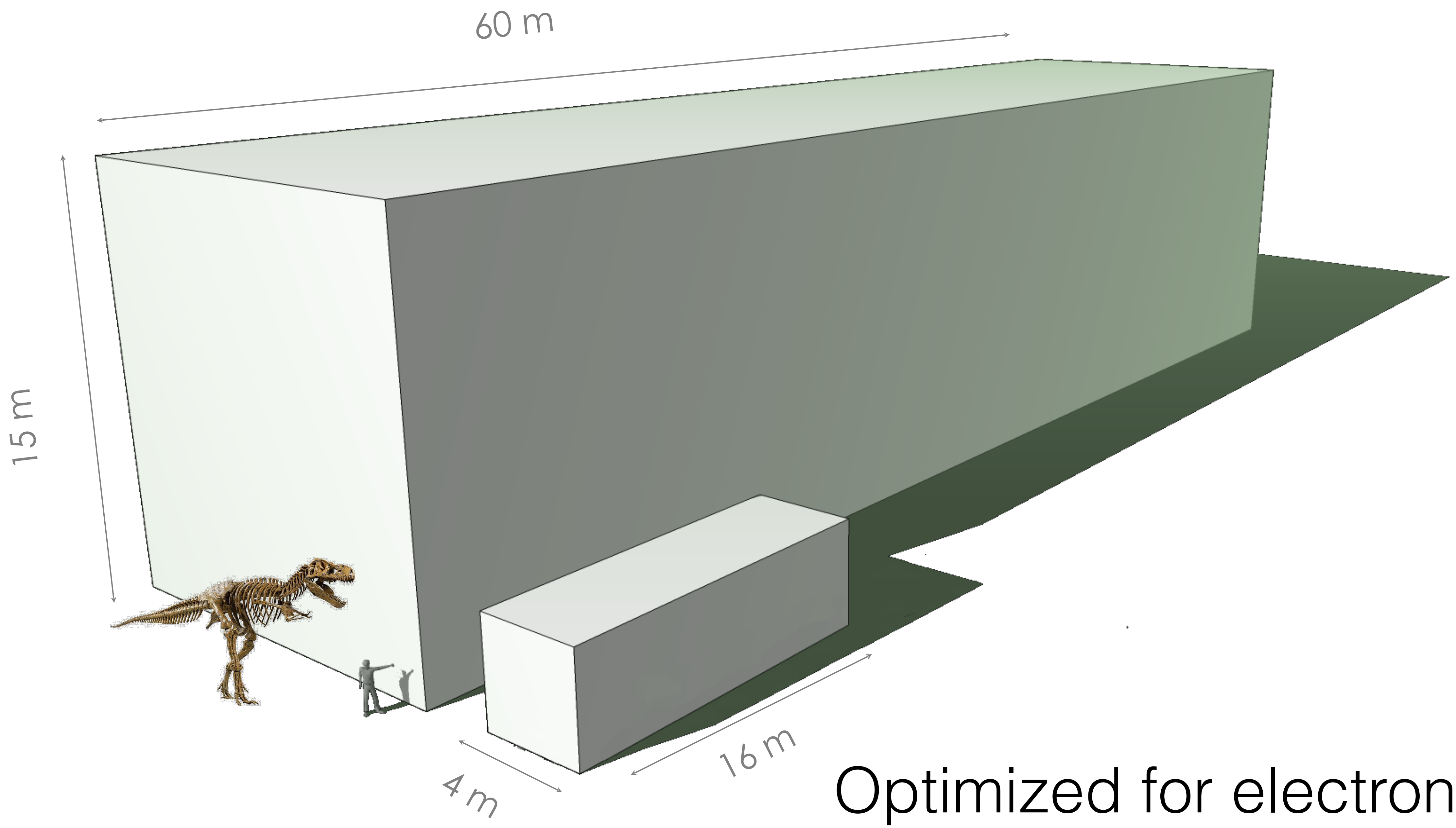






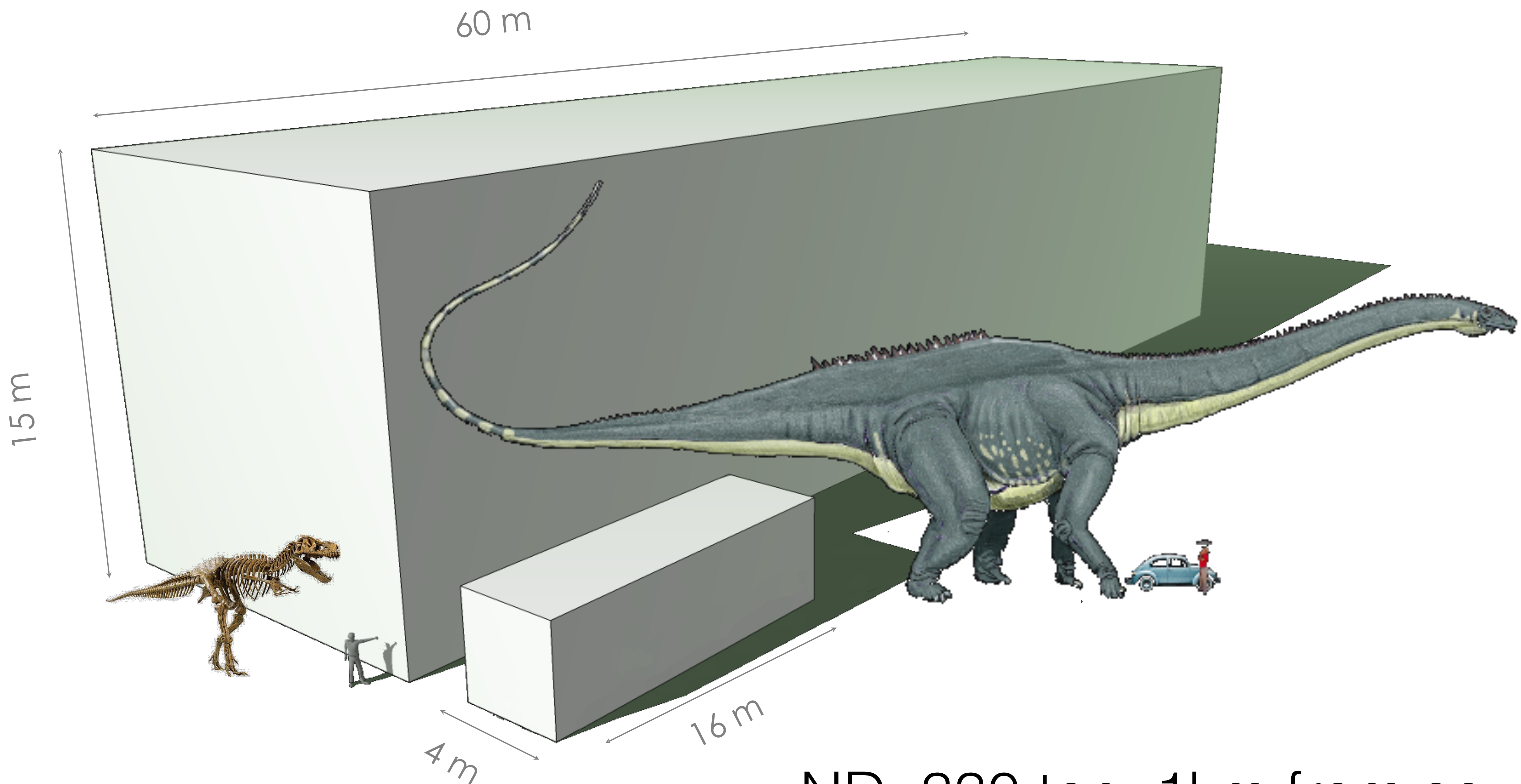


The NOvA Detectors



Optimized for electron ID, fine segmentation, Low-Z, and 65% Active

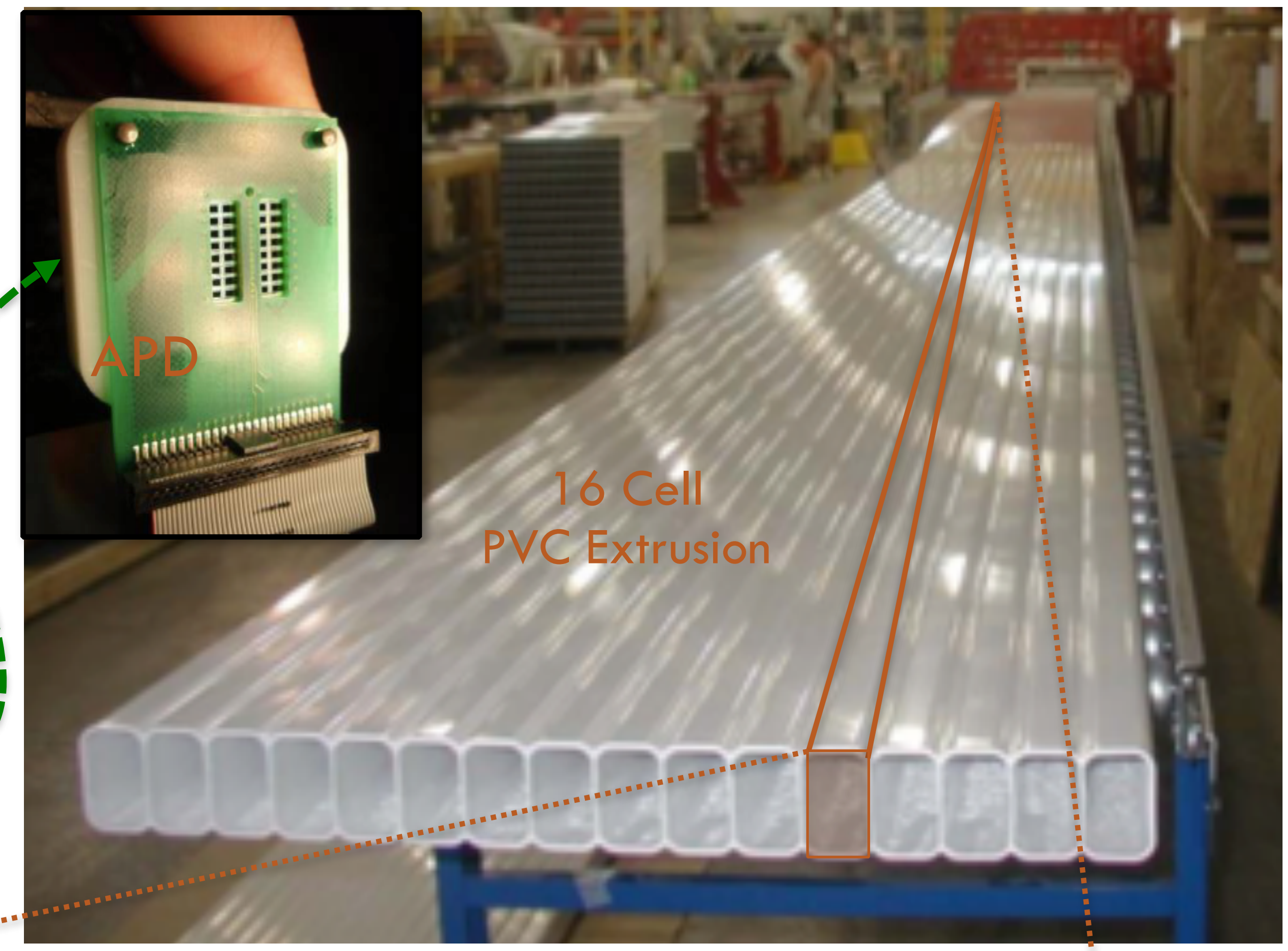
The NOvA Detectors

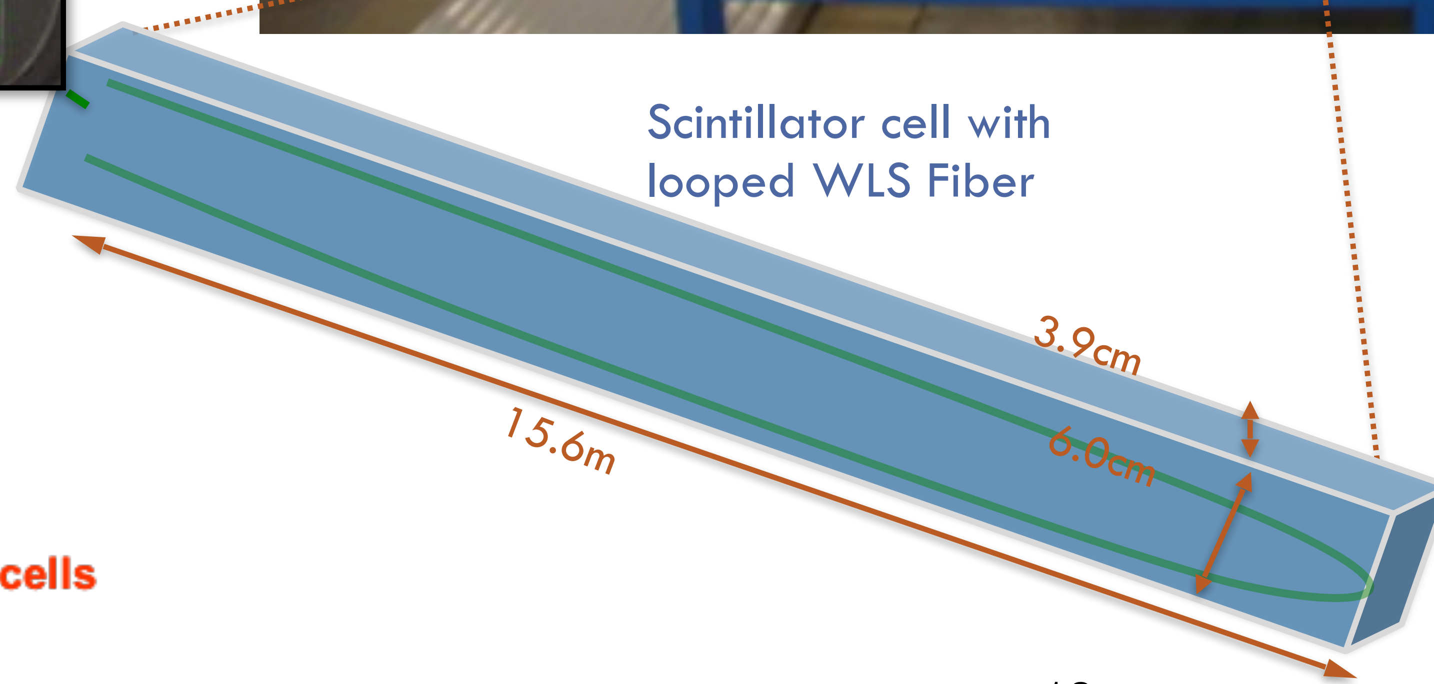


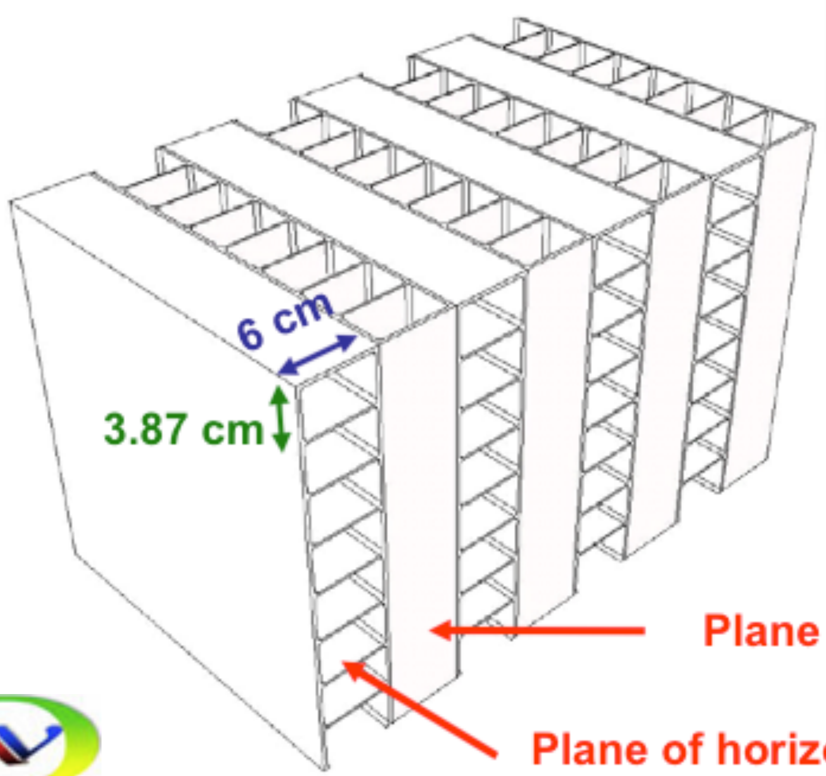
ND: 330 ton, 1km from source
FD: 14 kton, 810km from source

Detector Technology

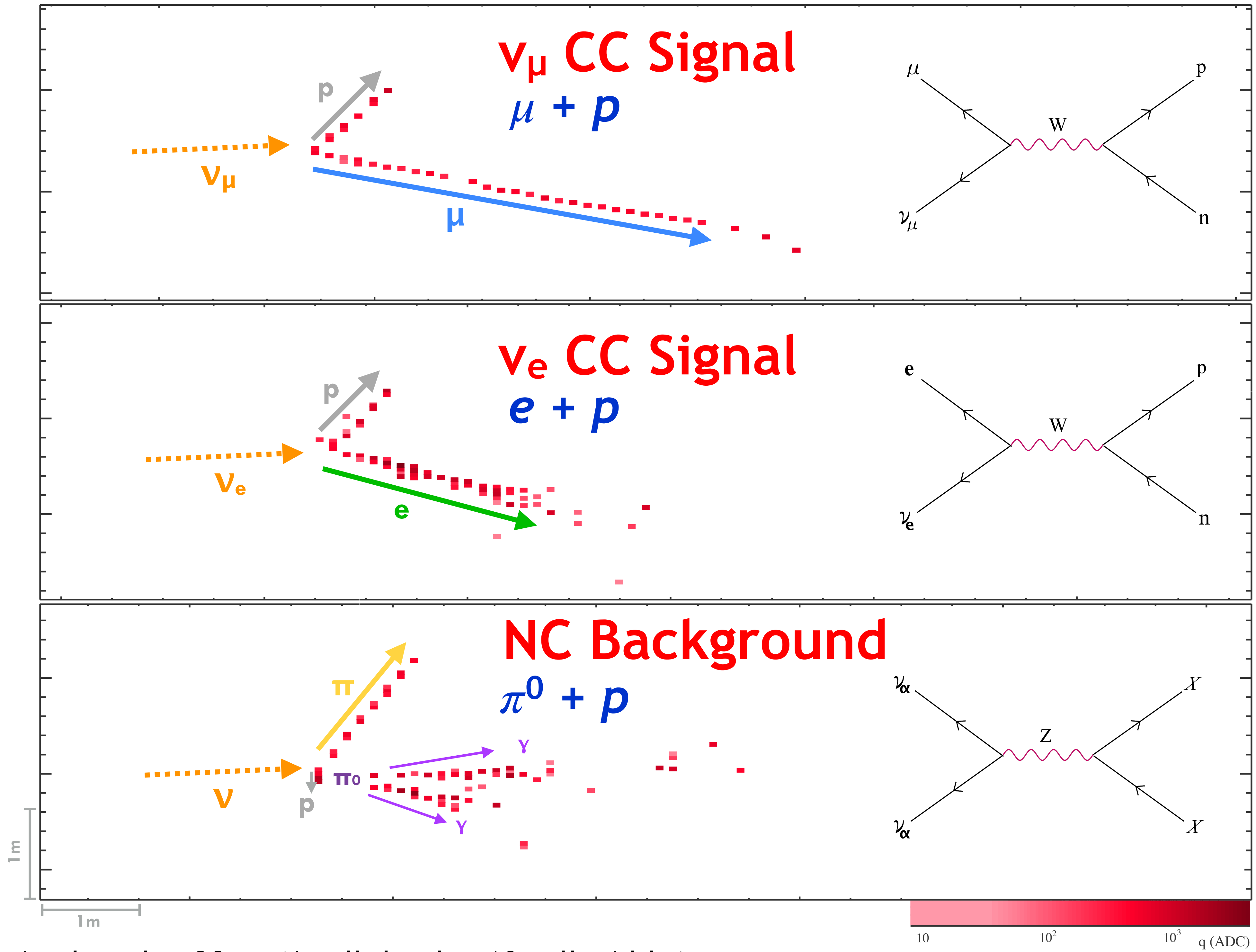
- PVC extrusion + Liquid Scintillator
  - mineral oil + 5% pseudocumene
- Read out via WLS fiber to APD
  - FD has 344,064 channels
  - muon crossing far end~25 PE
- Layered planes of orthogonal views





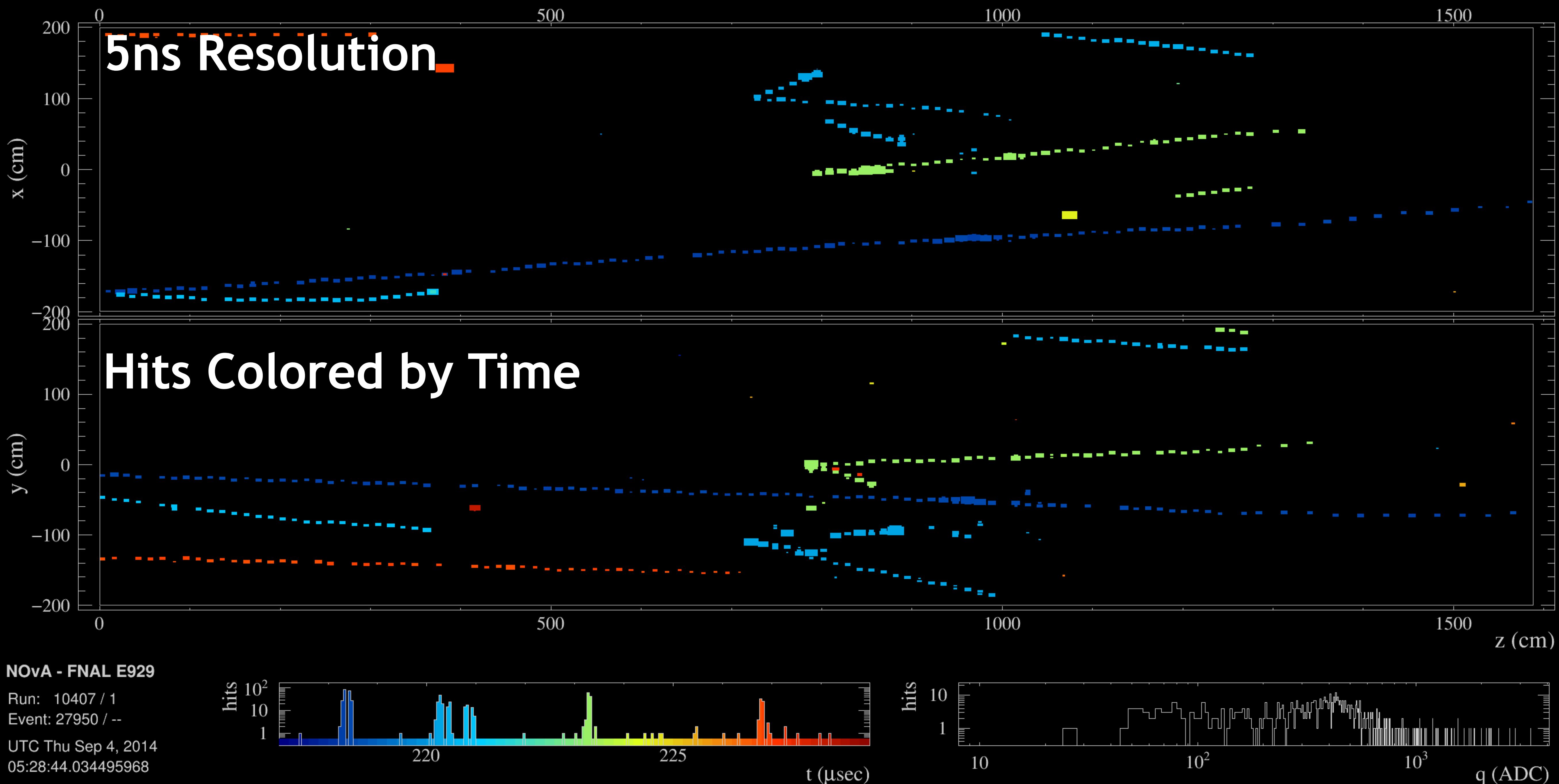


NOvA Event Topologies

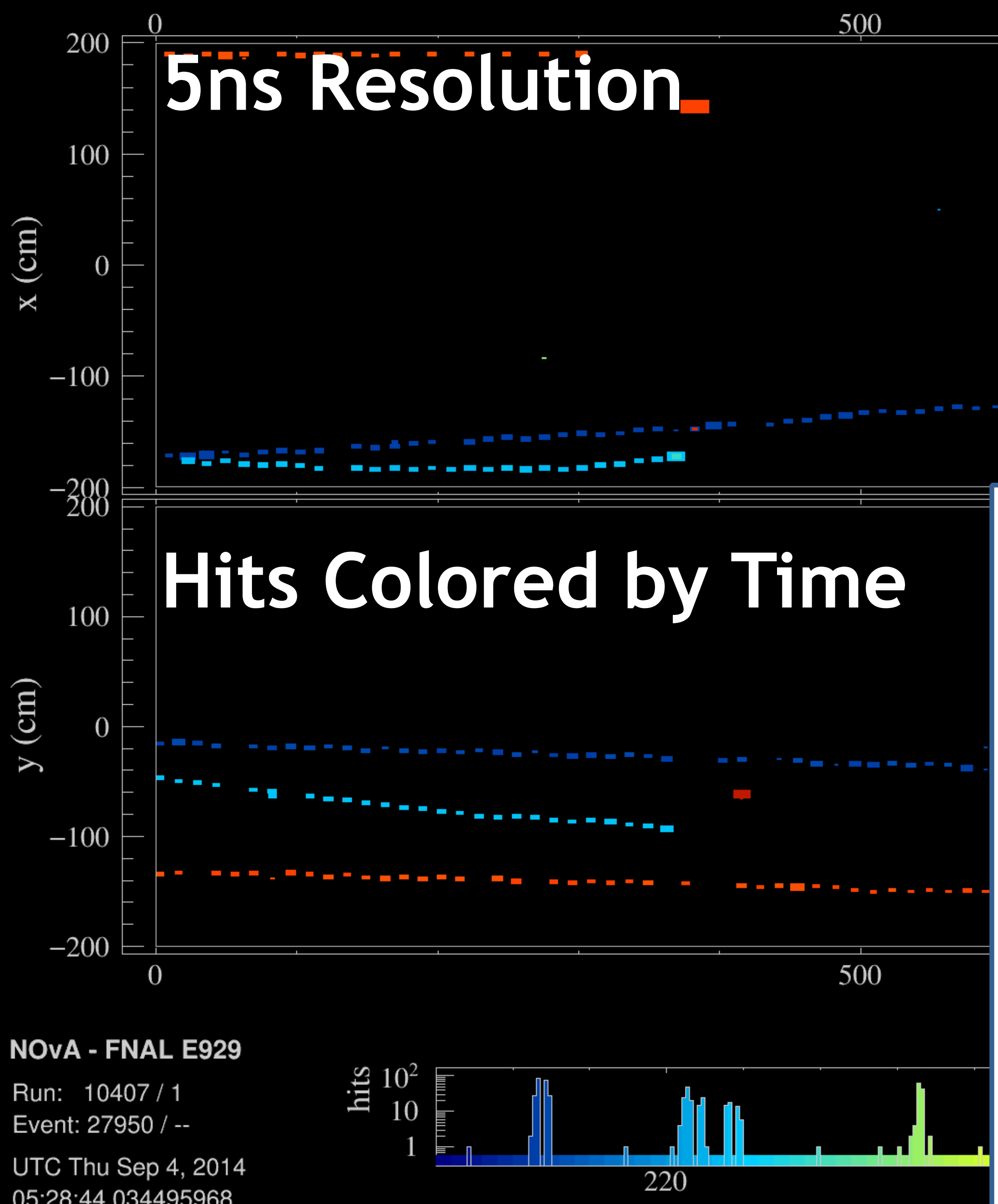


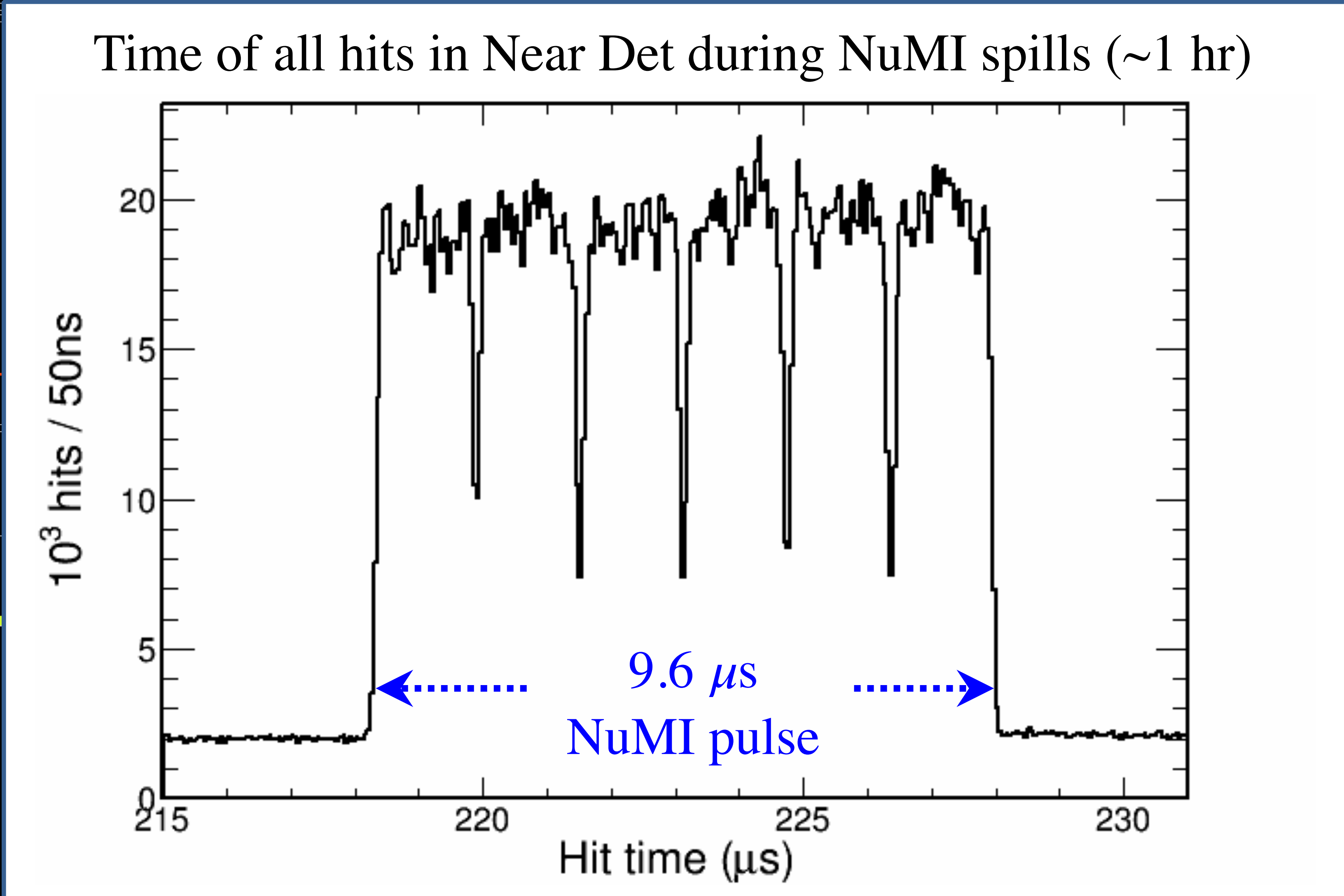
1 radiation length = 38cm (6 cell depths, 10 cell widths)

Near Detector: 10 μs of readout during NuMI beam pulse

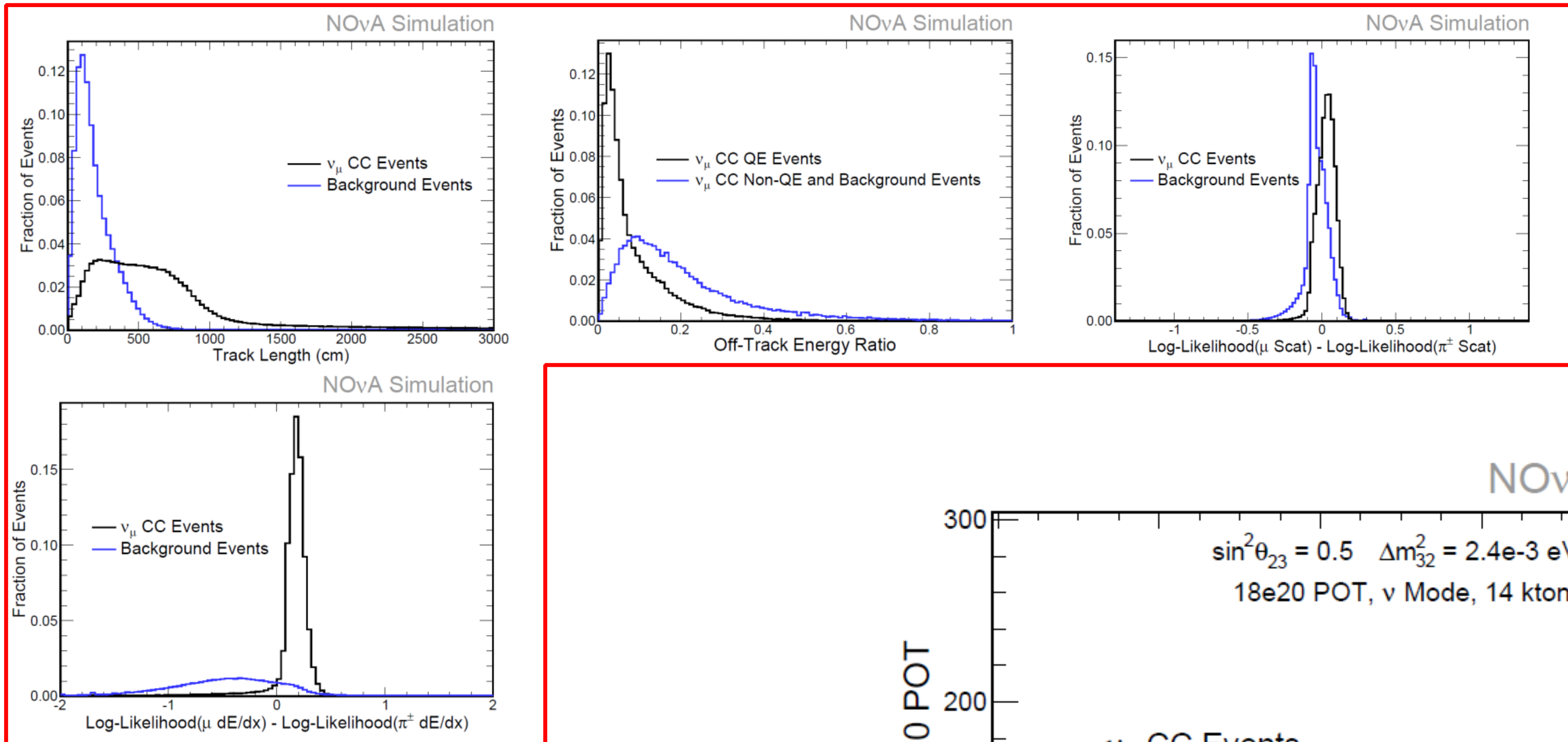


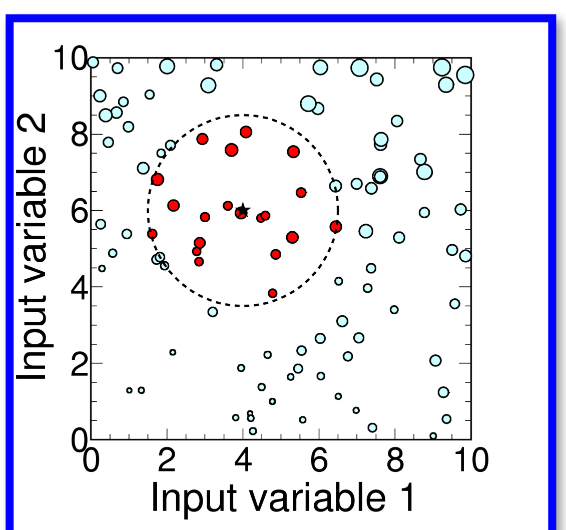
Near Detector: 10 μs of readout during NuMI beam pulse

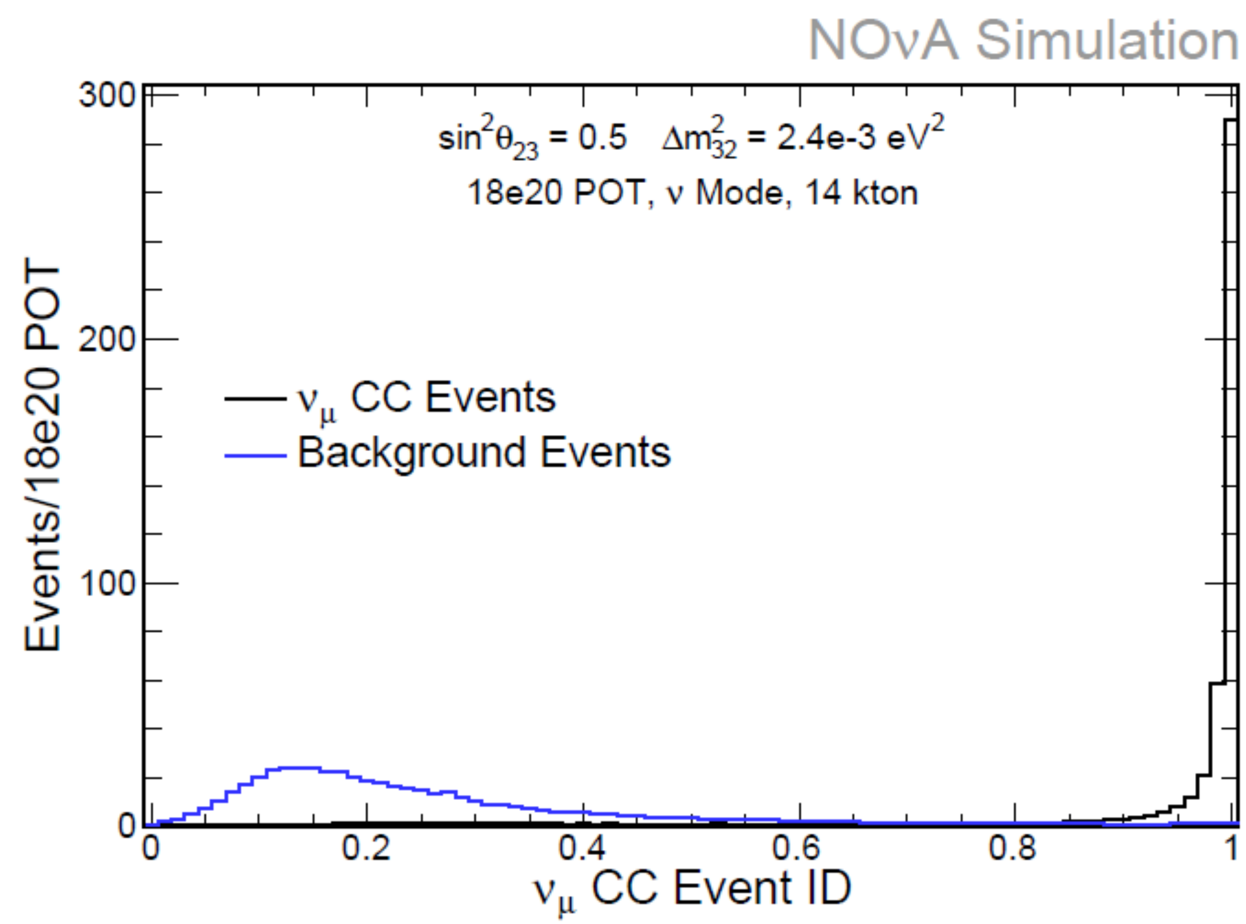




Conventional PIDs: ν_μ Selection





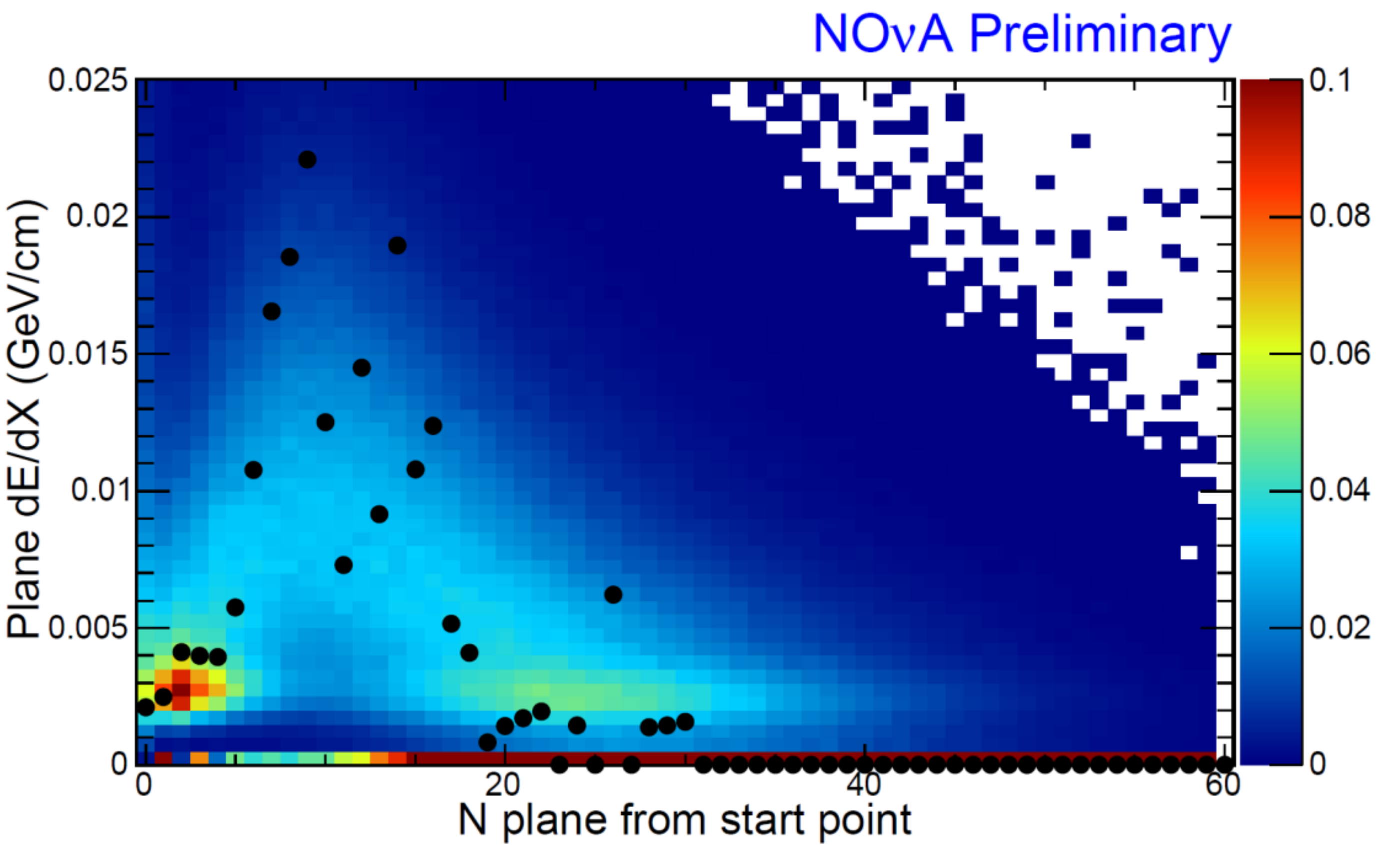


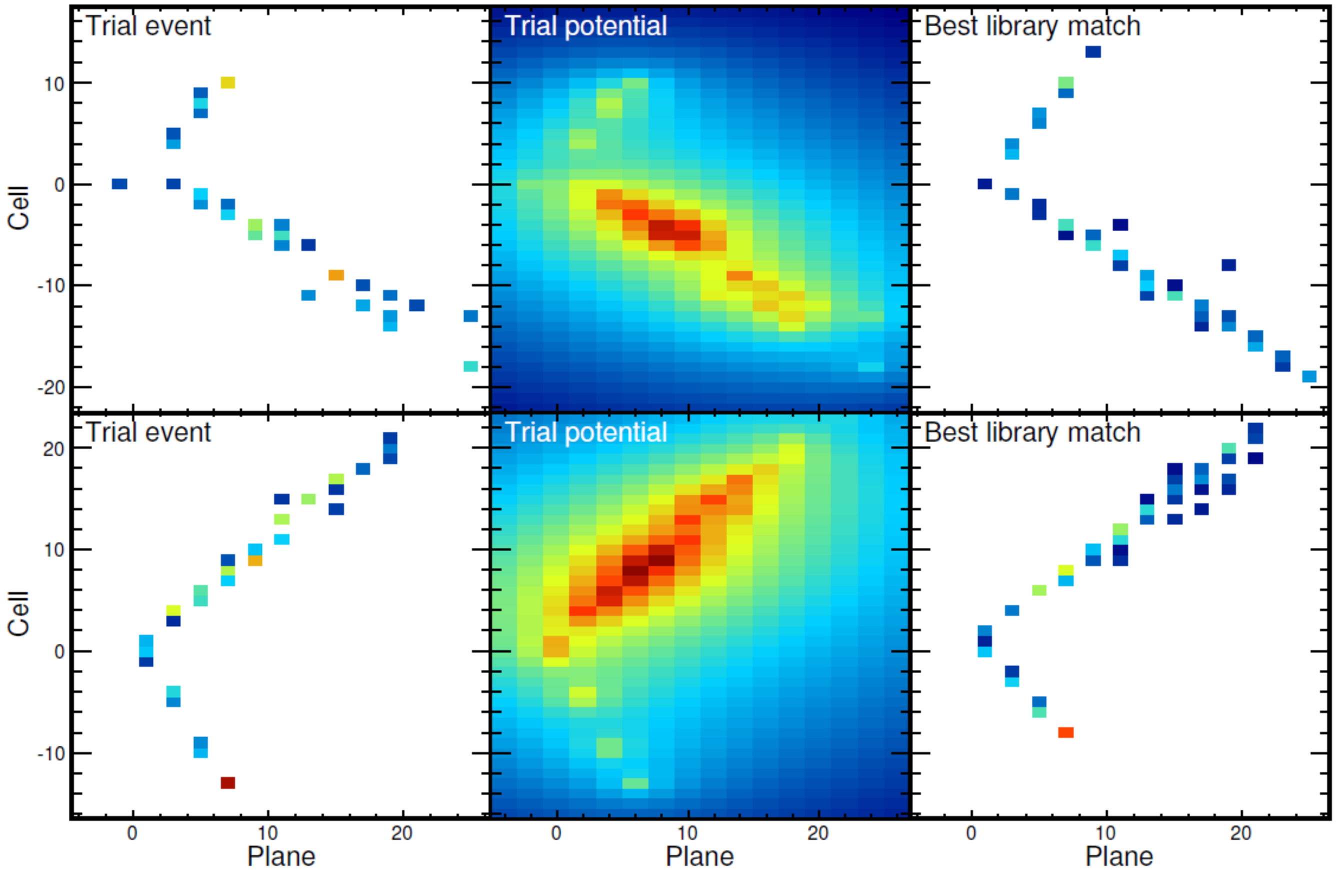
Conventional PIDs: ν_e Selection

LID:

Compares longitudinal and transverse de/dx in the leading shower to different particle hypotheses

Combines that with other topological information in an ANN to reject backgrounds





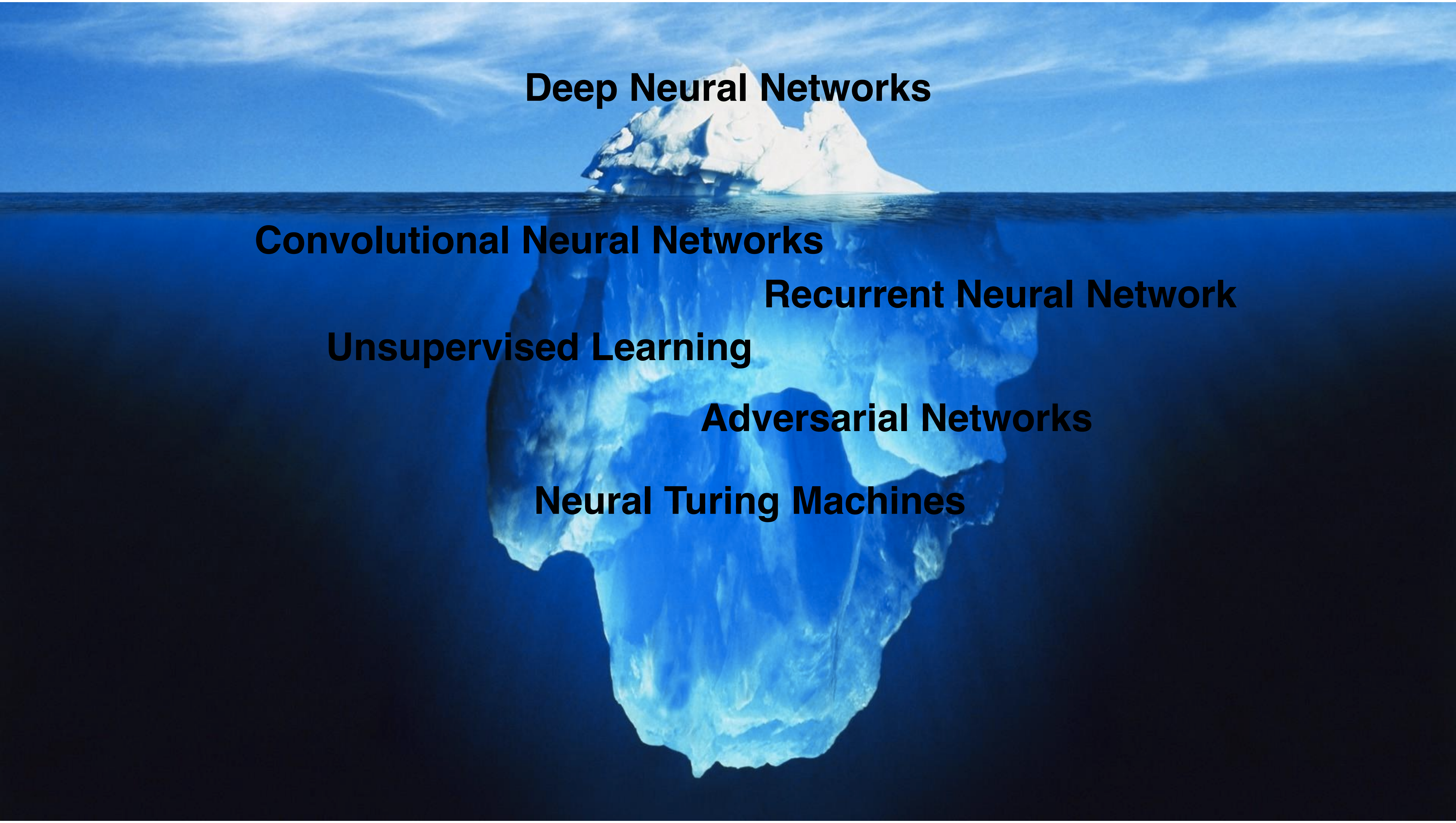
LEM:

Compares events to a vast library of simulated events

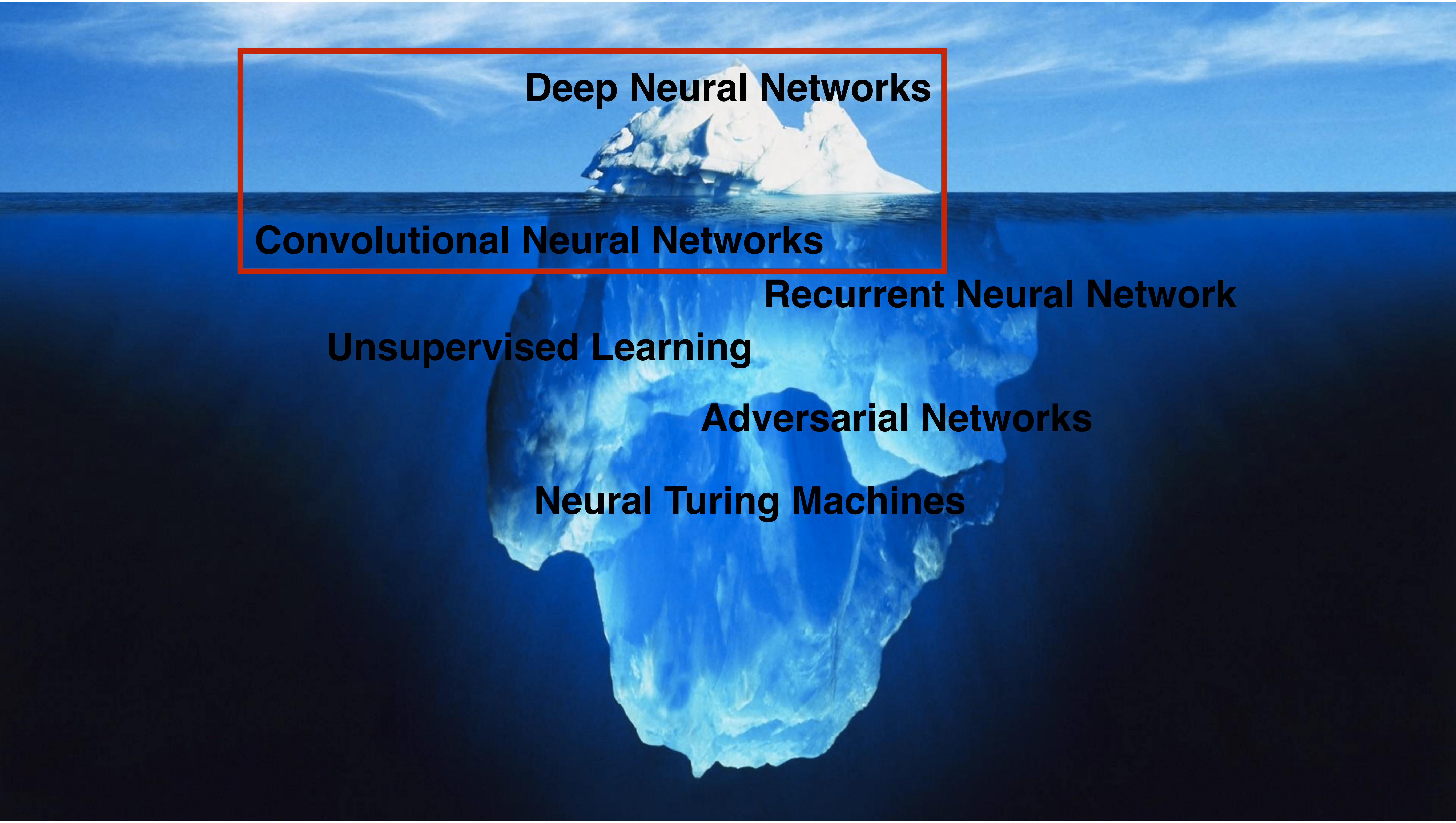
Information about the closest matches is fed into a boosted decision tree to reject backgrounds

Deep Learning

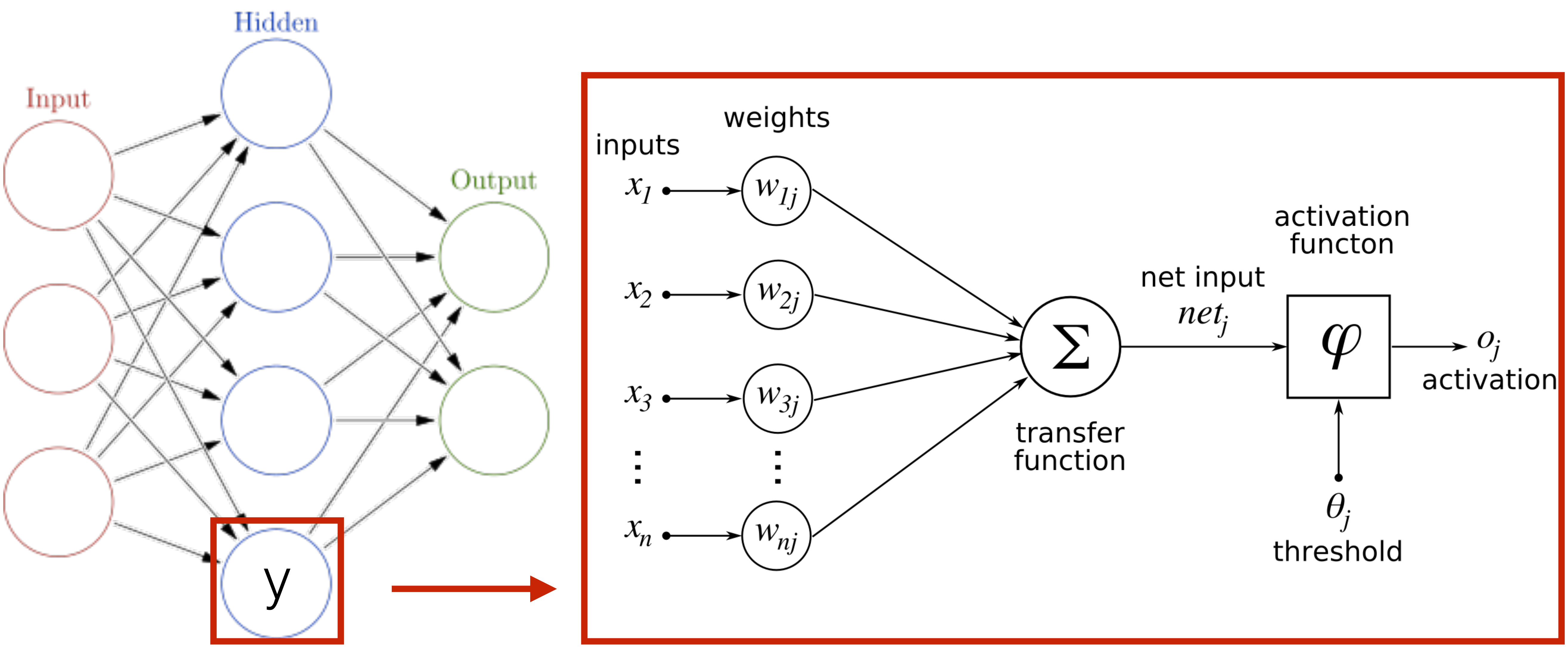
Deep Learning



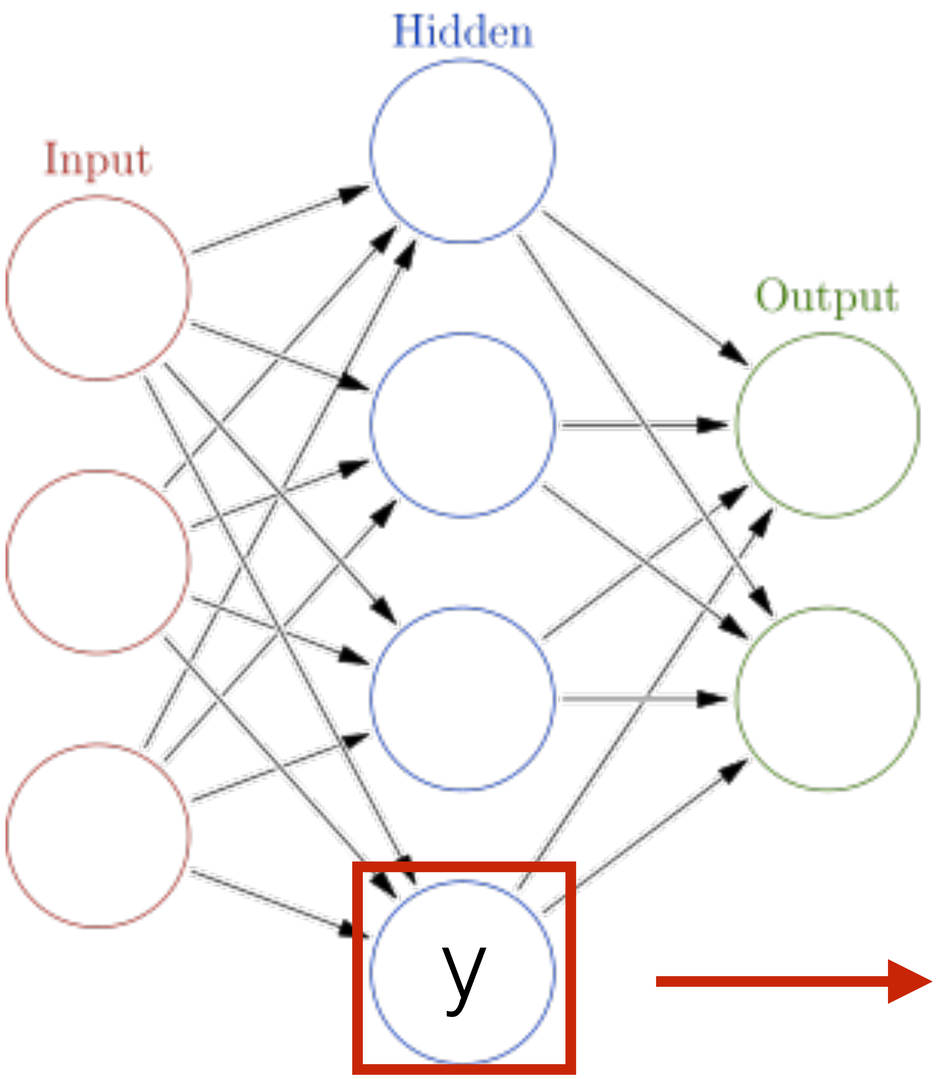
Deep Learning



Neural Networks



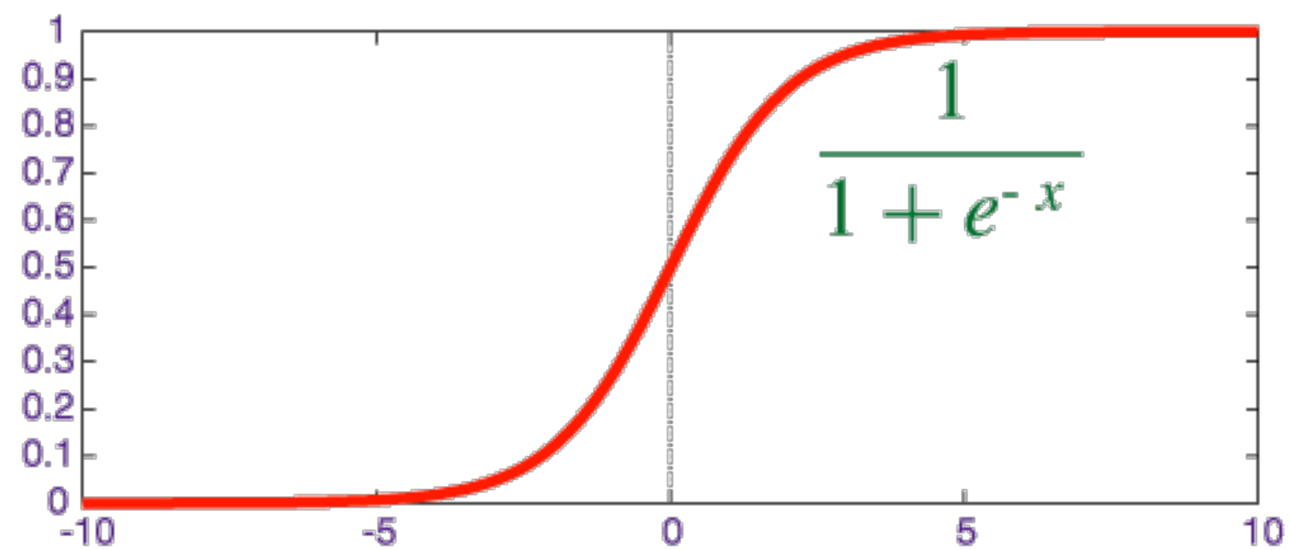
Neural Networks



$$x = input\ vector$$

$$y = \sigma\left(Wx + b\right)$$

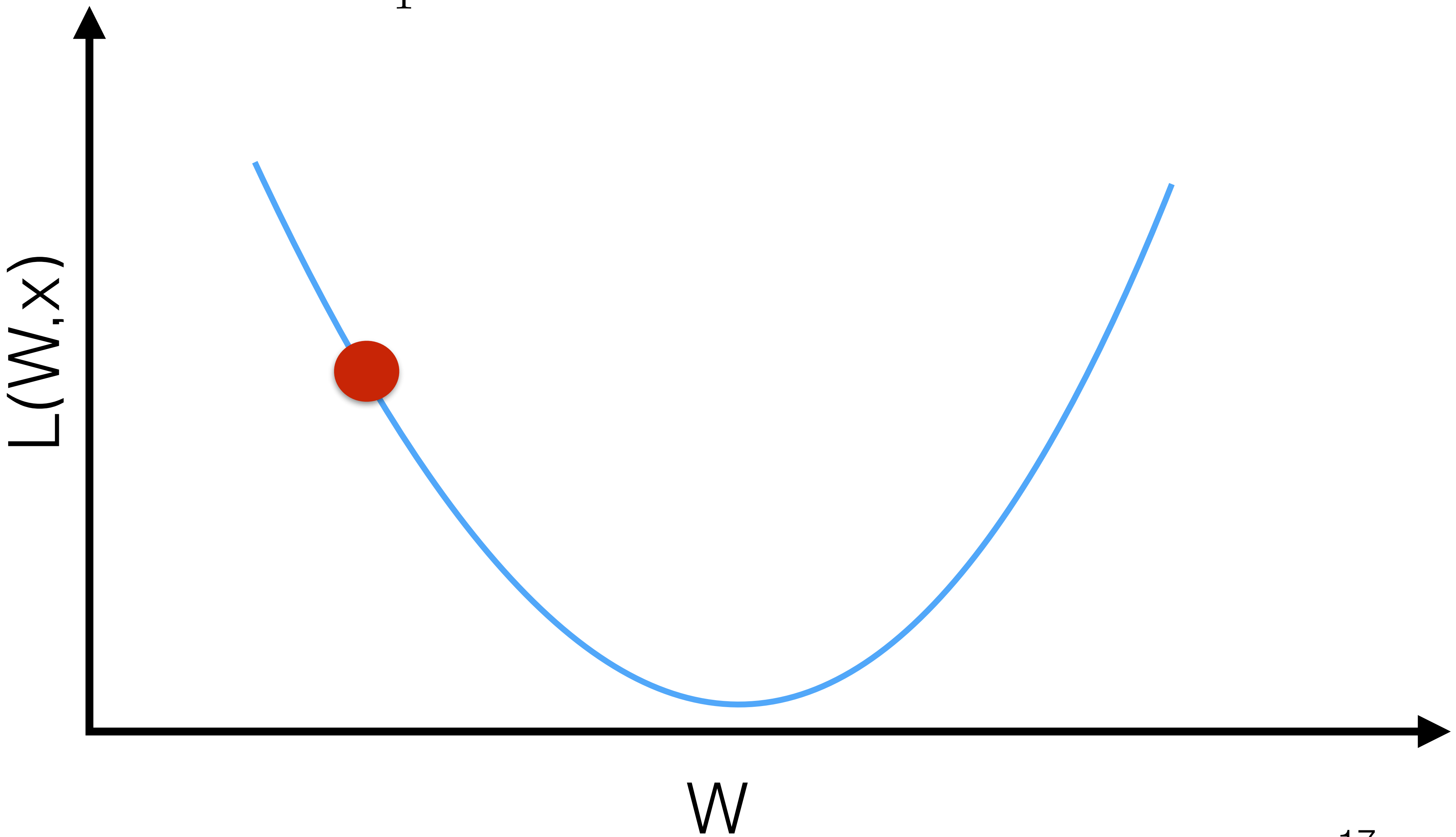
$$\sigma =$$



Training A Neural Network

Start with a "Loss" function which characterizes the performance of the network. For supervised learning:

$$L(W,X) = \frac{1}{N}\sum_{1}^{N_{examples}} -y_i \log\left(f(x_i)\right) - (1-y_i)\log\left(1-f(x_i)\right)$$



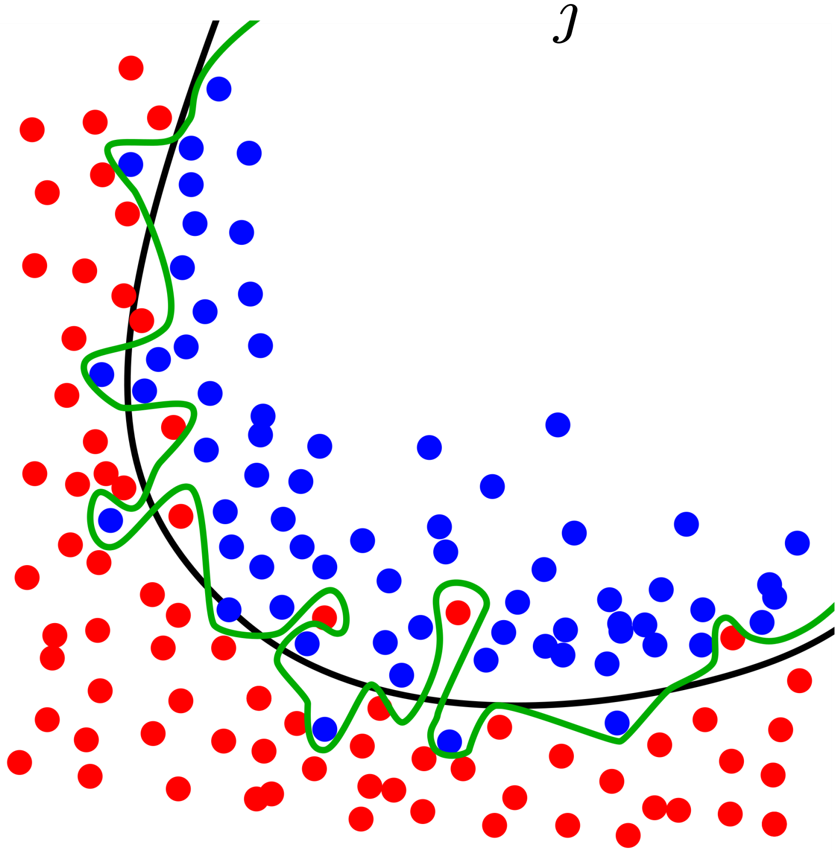
Training A Neural Network

Start with a “Loss” function which characterizes the performance of the network. For supervised learning:

$$L(W,X) = \frac{1}{N}\sum_{1}^{N_{examples}} -y_i \log\left(f(x_i)\right) - (1-y_i)\log\left(1-f(x_i)\right)$$

Add in a regularization term to avoid overfitting:

$$L' = L + \frac{1}{2}\sum_j w_j^2$$



Training A Neural Network

Start with a "Loss" function which characterizes the performance of the network. For supervised learning:

$$L(W,X) = \frac{1}{N}\sum_{1}^{N_{examples}} -y_i \log\left(f(x_i)\right) - (1-y_i)\log\left(1-f(x_i)\right)$$

Add in a regularization term to avoid overfitting:

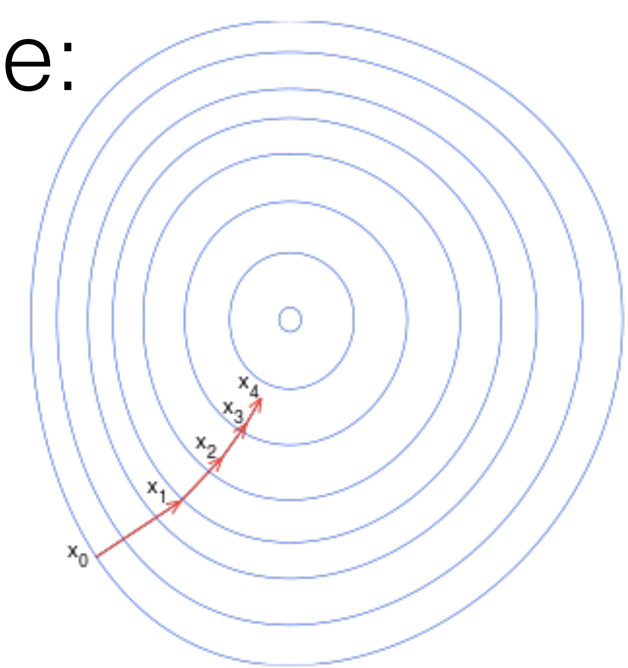
$$L' = L + \frac{1}{2}\sum_j w_j^2$$

Propagate the gradient of the network back to specific nodes using back propagation. AKA apply the chain rule:

$$\nabla_{w_j} L = \frac{\delta L}{\delta f}\frac{\delta f}{\delta g_n}\frac{\delta g_n}{\delta g_{n-1}}\cdots\frac{\delta g_{k+1}}{\delta g_k}\frac{\delta g_k}{\delta w_j}$$

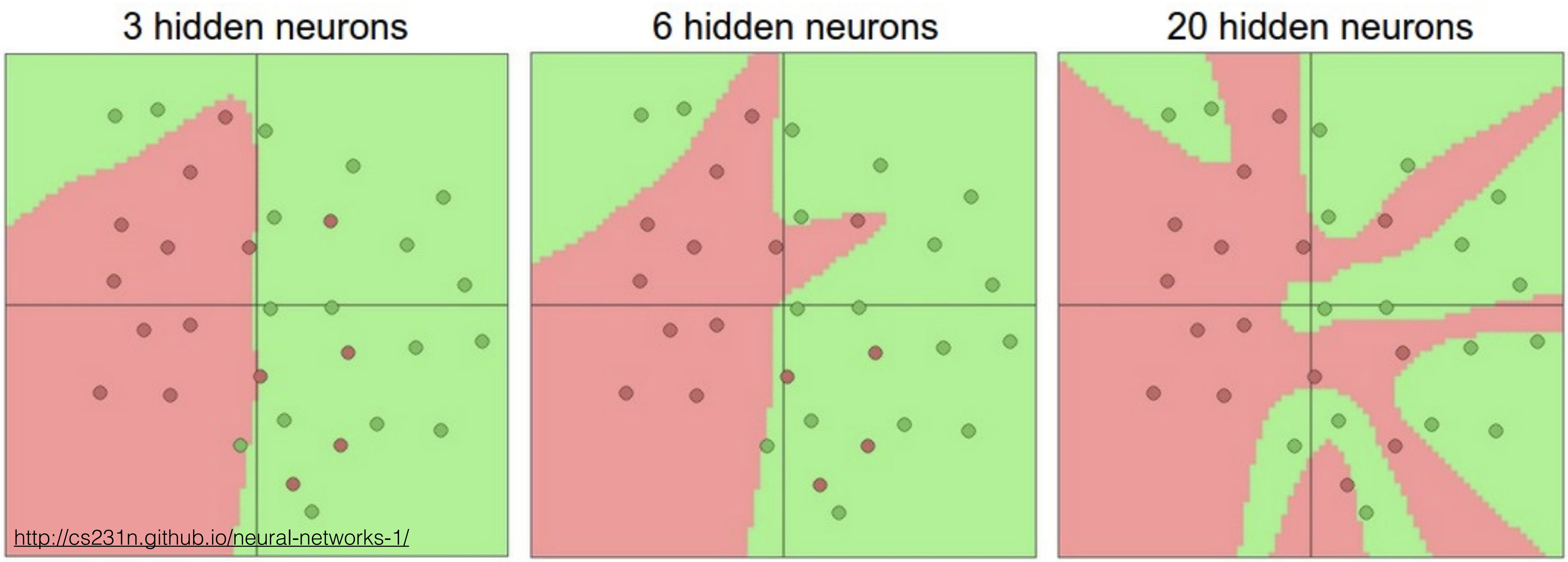
Update weights using gradient descent:

$$w_j' = w_j - \alpha \nabla_{w_j} L$$



Deep Neural Networks

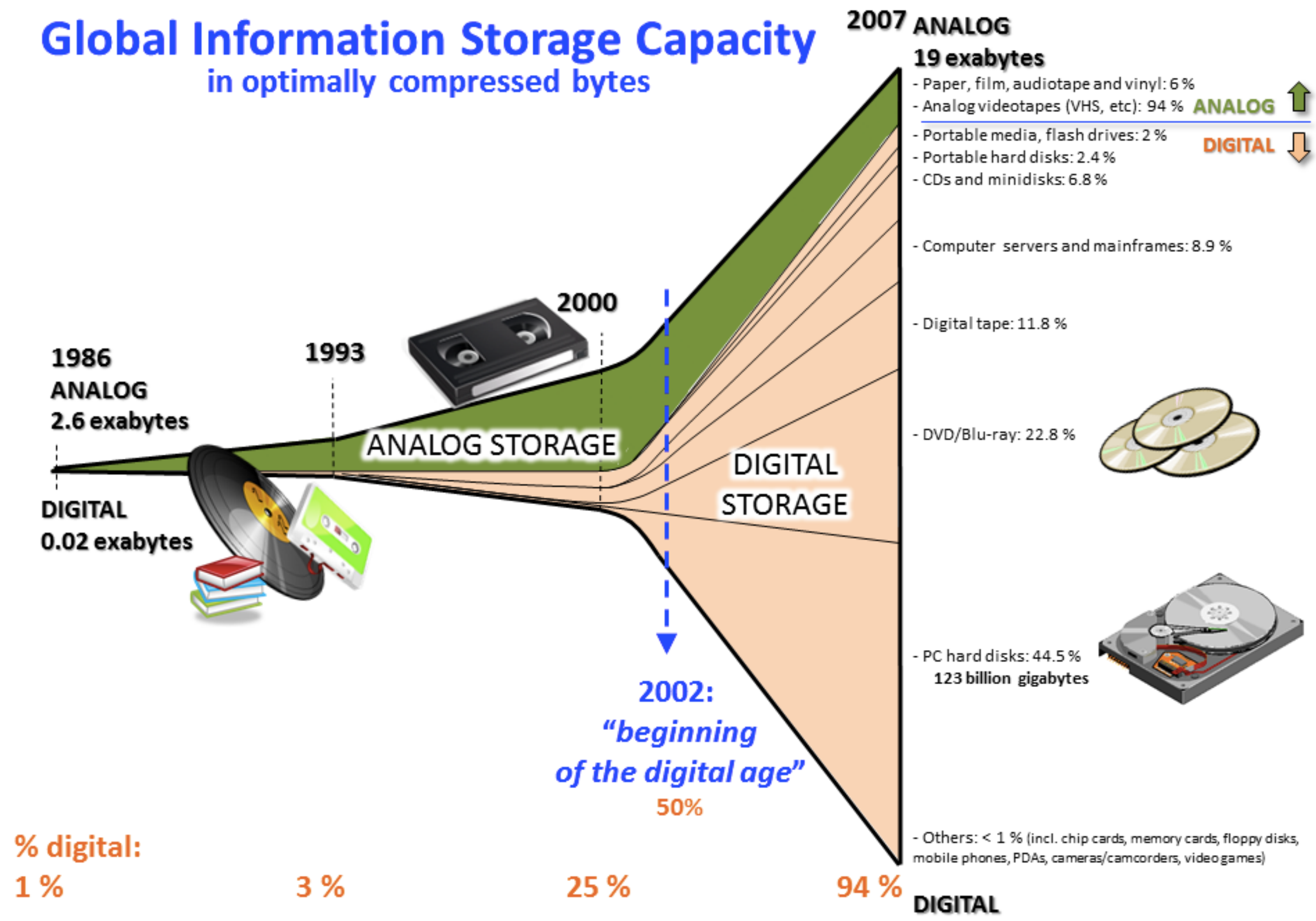
What if we try to keep all the input data? Why not rely on a wide, extremely Deep Neural Network (DNN) to learn the features it needs? Sufficiently deep networks make excellent function approximators:



However, until recently they proved almost impossible to train.

The Deep Learning Revolution

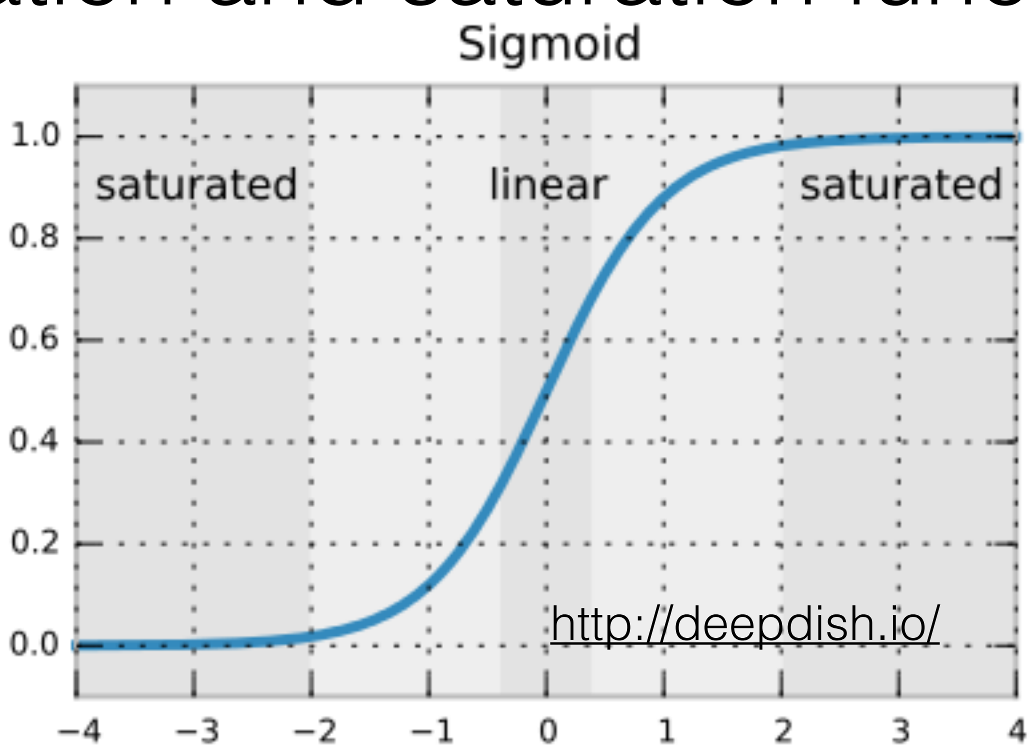
Most famously "big data" and accelerated computing made it easier to find and run deep neural networks:

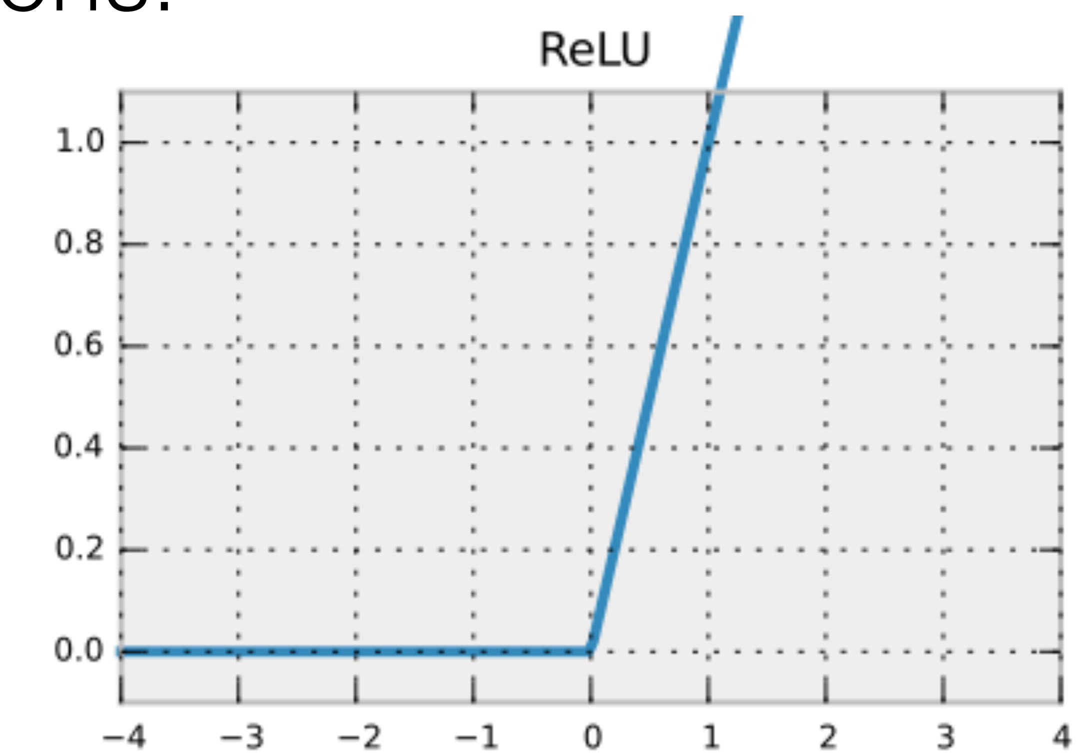


Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

Better Activation Functions

But there were also some major technical breakthroughs. One being more effective back propagation due to better weight initialization and saturation functions:





The problem with sigmoids:

$$\frac{\delta\sigma(x)}{\delta x} = \sigma(x)\left(1 - \sigma(x)\right)$$

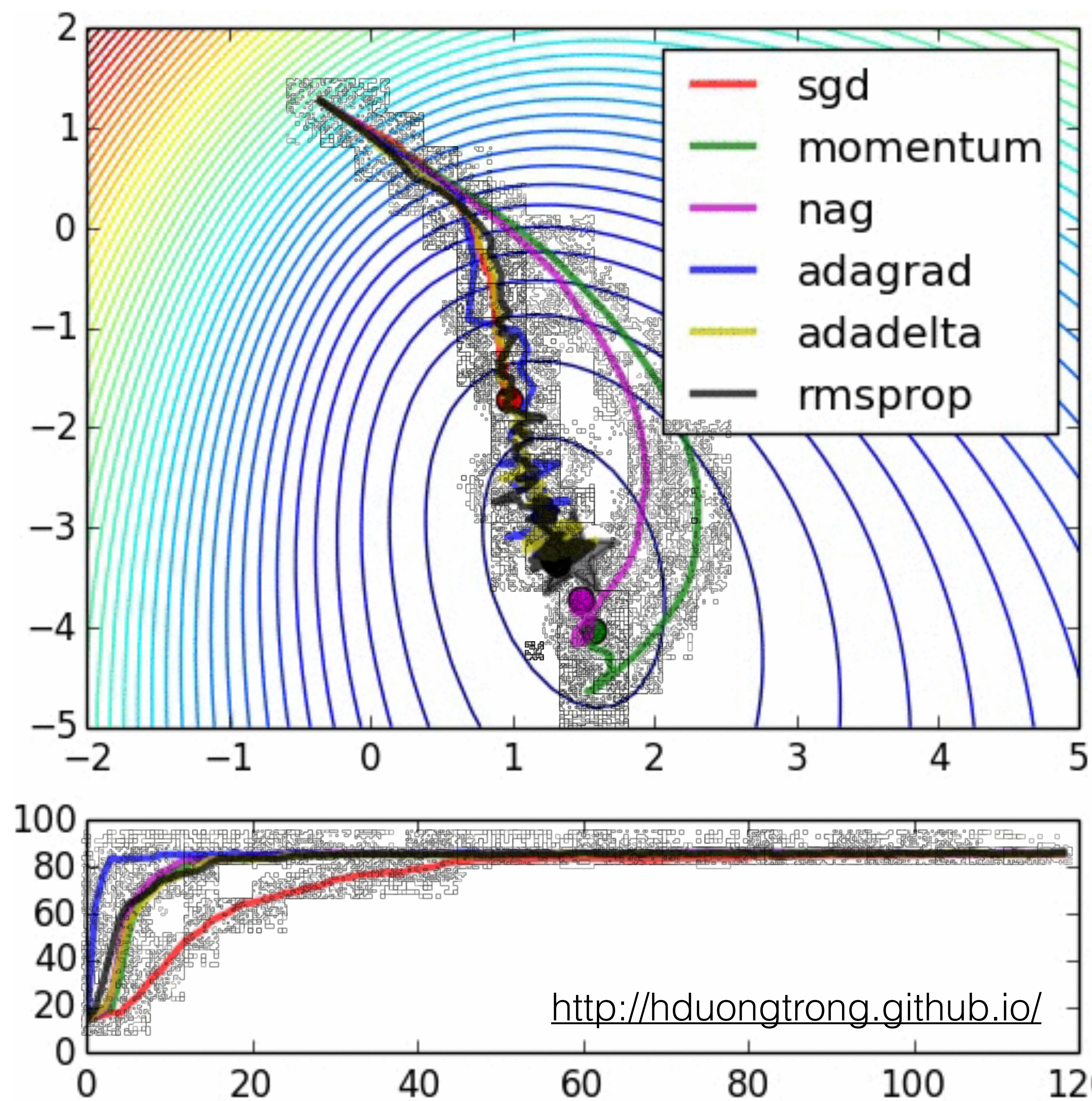
ReLU:

$$\frac{ReLU(x)}{\delta x} = \begin{cases} 1 & \text{when } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Sigmoid gradient goes to 0 when x is far from 1. Makes back propagation impossible! Use ReLU to avoid saturation.

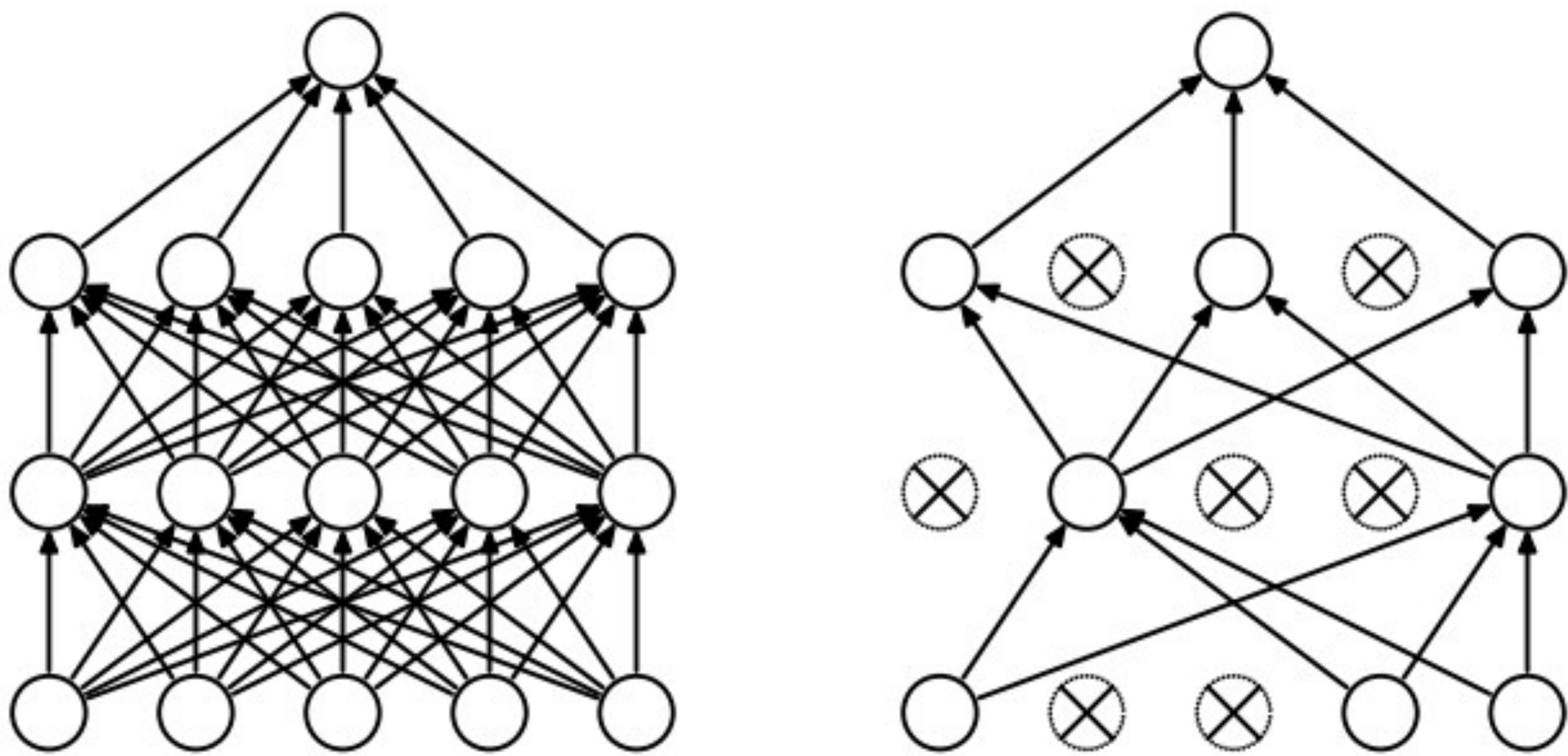
Smarter Training

Another is stochastic gradient descent (SGD). In SGD we avoid some of the cost of gradient descent by evaluating as few as one event at a time. The performance of conventional gradient descent is approximated as the various noisy sub estimates even out, with the stochastic behavior even allowing for jumping out of local minima.



Dropout

- Same goal as conventional regularization- prevent overtraining.
- Works by randomly removing whole nodes during training iterations. At each iteration, randomly set XX% of weights to zero and scale the rest up by 1/(1 – 0.XX).
- Forces the network not to build complex interdependencies in the extracted features.

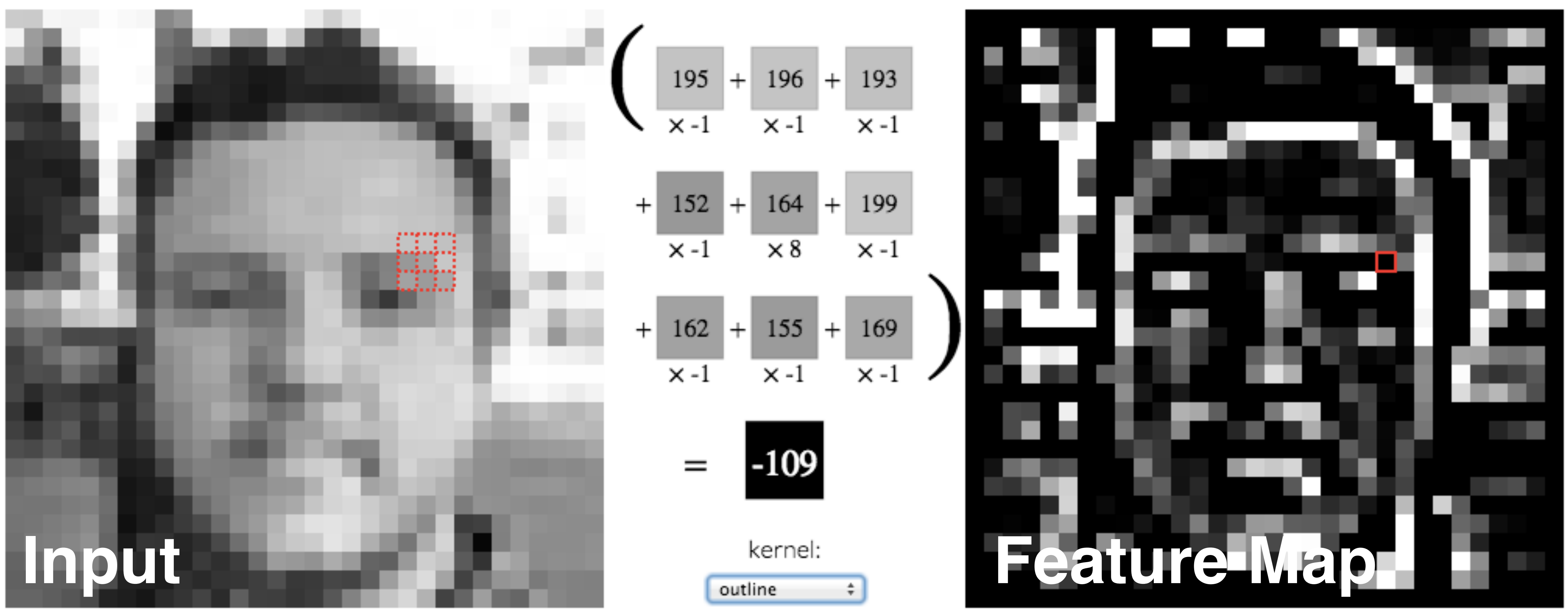


(a) Standard Neural Net (b) After applying dropout.

Convolutional Neural Networks

Instead of training a weight for every input pixel, try learning weights that describe kernel operations, convolving that kernel across the entire image to exaggerate useful features.

Inspired by research showing that cells in the visual cortex are only responsive to small portions of the visual field.

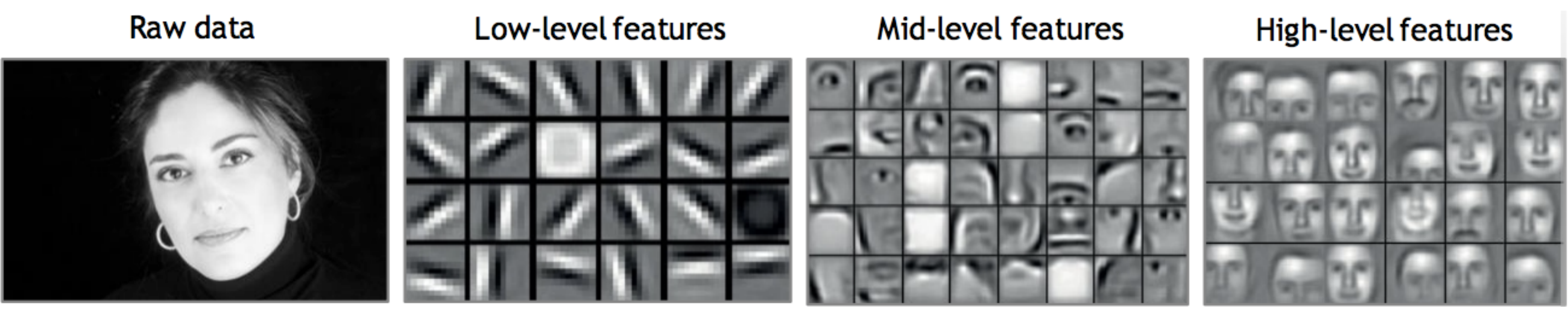


Kernel

http://setosa.io/ev/image-kernels/

Convolutional Neural Networks

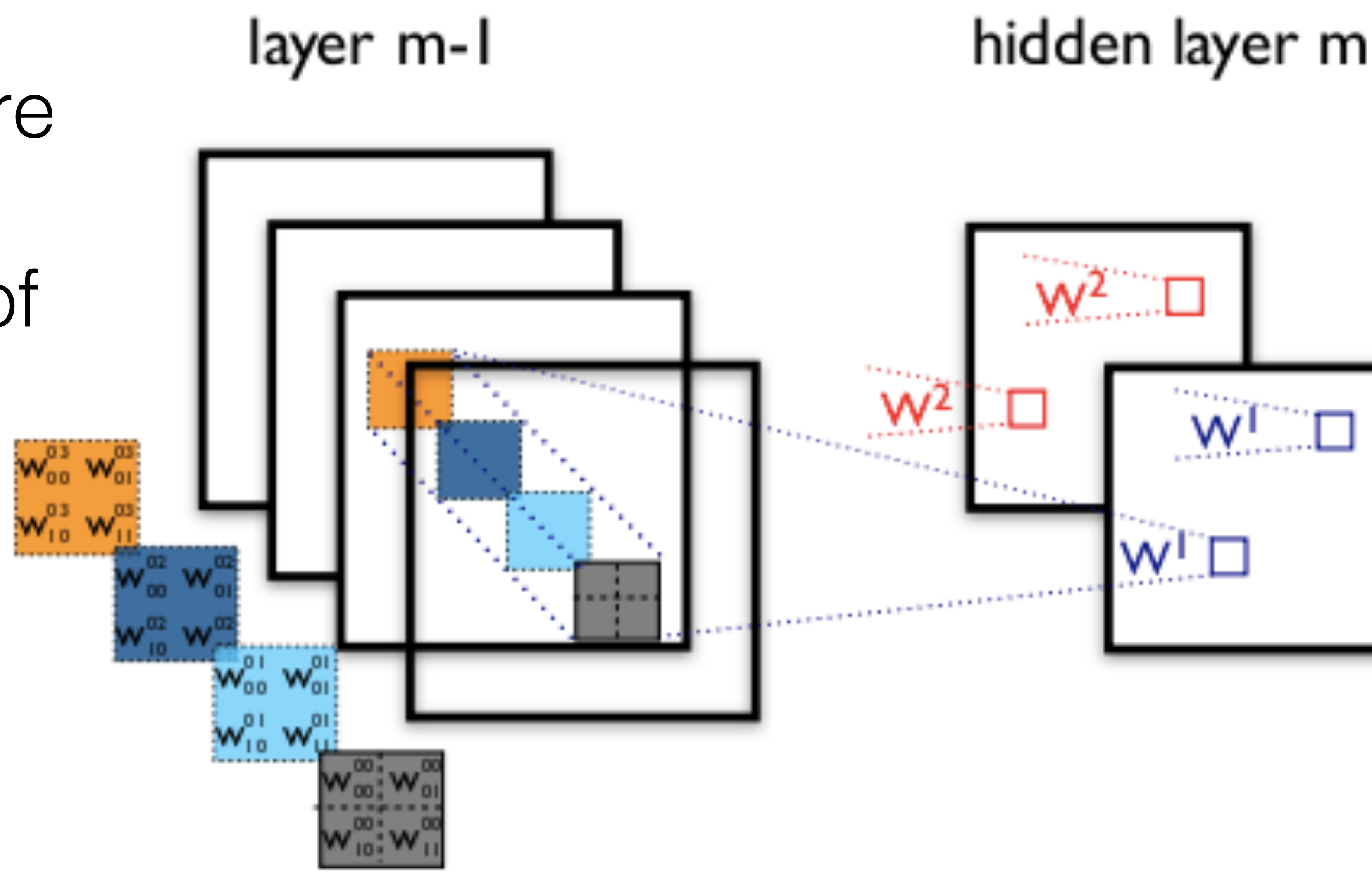
Instead of training a weight for every input pixel, try learning weights that describe kernel operations, convolving that kernel across the entire image to exaggerate useful features. Inspired by research showing that cells in the visual cortex are only responsive to small portions of the visual field.



https://developer.nvidia.com/deep-learning-courses

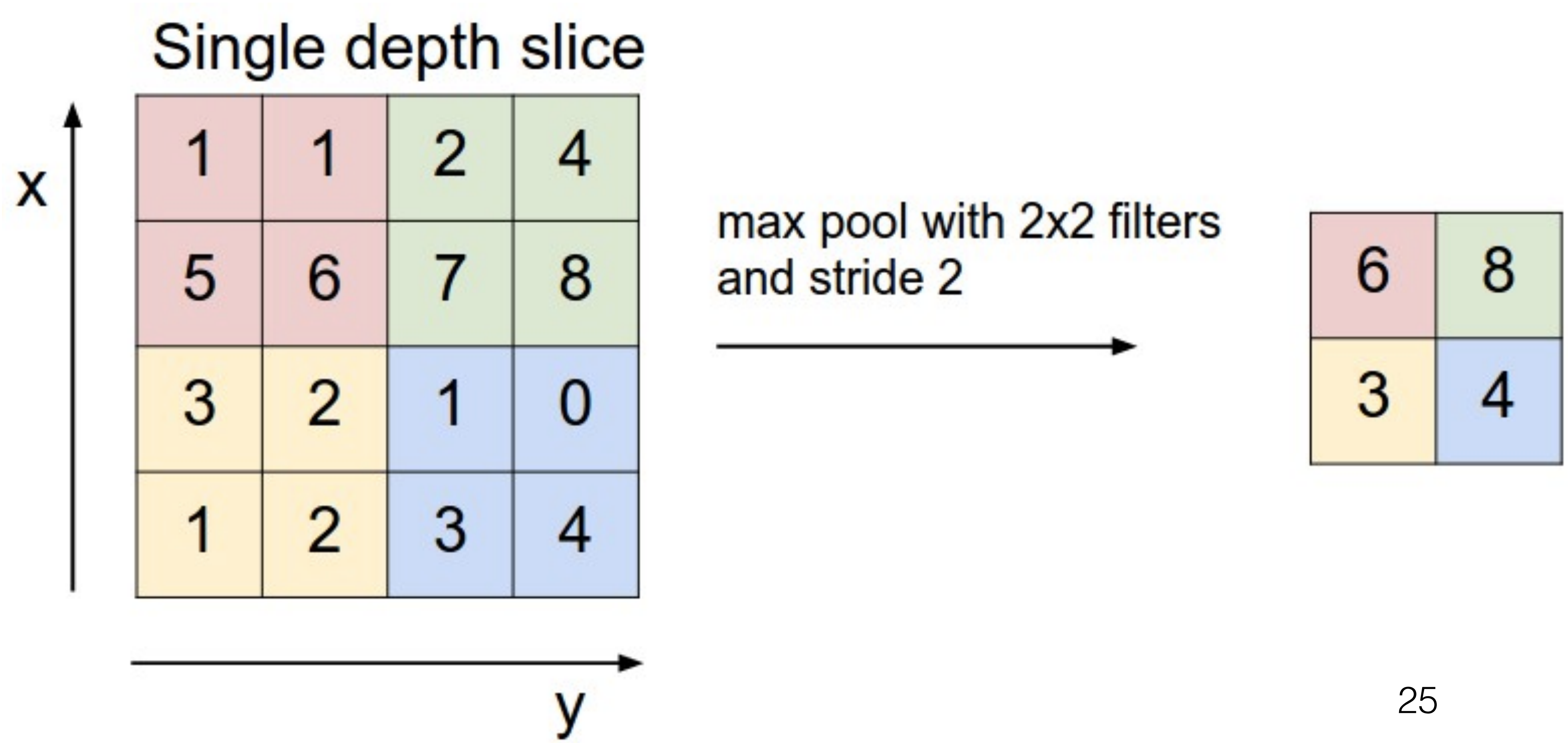
Convolutional Layers

- Every trained kernel operation is the same across an entire input image or feature map.
- Each convolutional layer trains an array of kernels to produce output feature maps.
- Weights for a given convolutional layer are a 4D tensor of NxMxHxW (number of incoming features, number of outgoing features, height, and width)



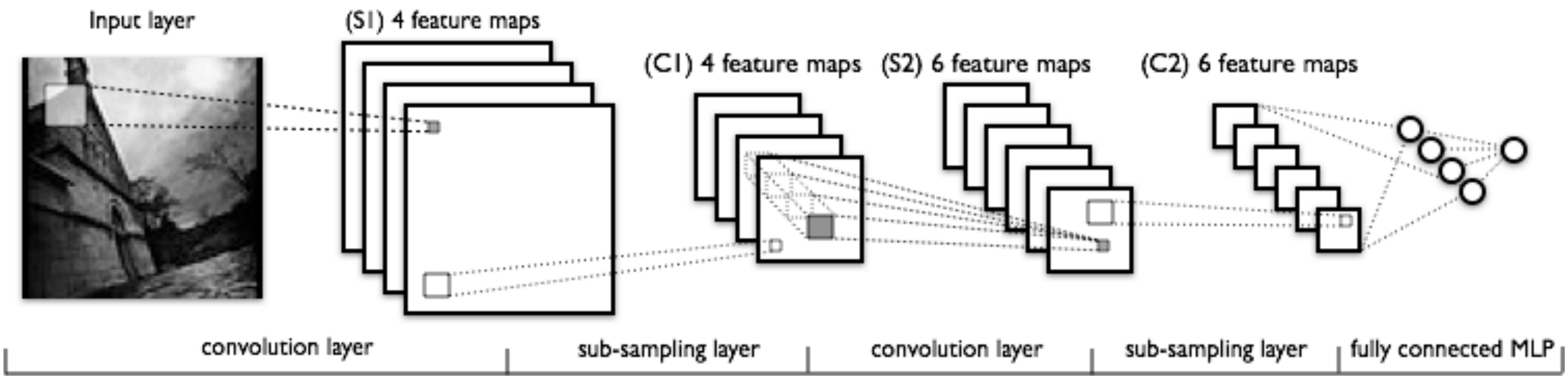
Pooling Layers

- Intelligent downscaling of input feature maps.
- Stride across images taking either the maximum or average value in a patch.
- Same number of feature maps, with each individual feature map shrunk by an amount dependent on the stride of the pooling layers.



The LeNet

In its simplest form a convolutional neural network is a series of convolutional, max pooling, and MLP layers:

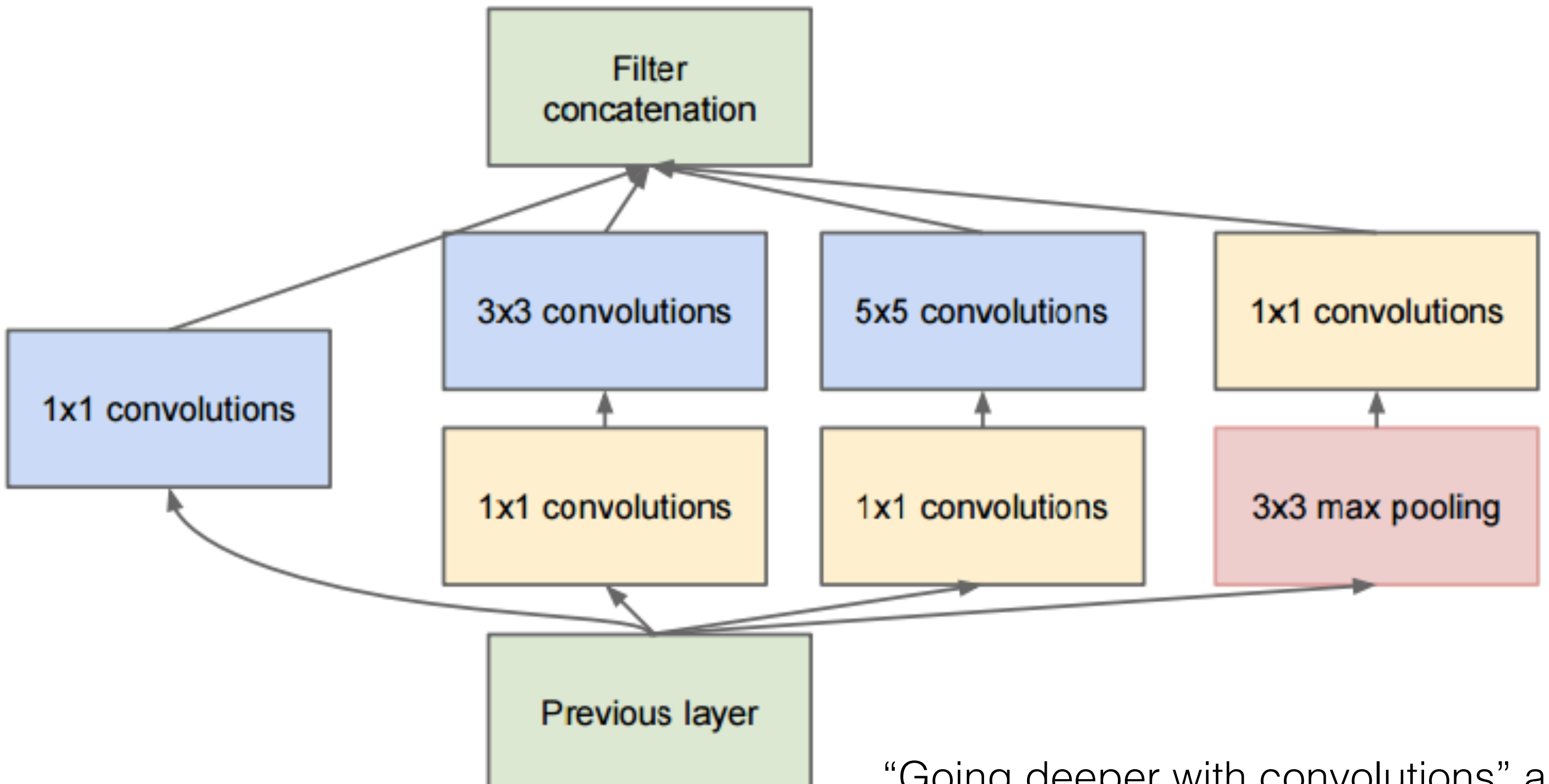


The "LeNet" circa 1989

http://deeplearning.net/tutorial/lenet.html http://yann.lecun.com/exdb/lenet/

Modern CNNs

Renaissance in CNN use over the last few years, with increasingly complex network-in-network models that allow for deeper learning of more complex features.



“Going deeper with convolutions” arXiv:1409.4842

The brilliance of this inception module is that it uses kernels of several sizes but keeps the number of feature maps under control by use of a 1x1 convolution.

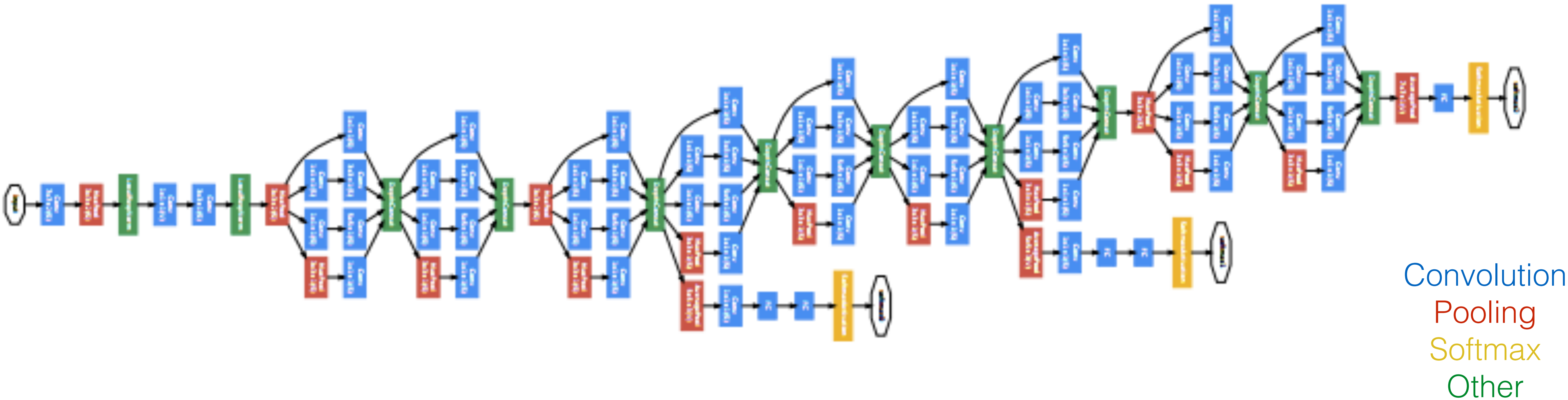
Modern CNNs

Renaissance in CNN use over the last few years, with increasingly complex network-in-network models that allow for deeper learning of more complex features.

The brilliance of this inception module is that it uses kernels of several sizes but keeps the number of feature maps under control by use of a 1x1 convolution.

Modern CNNs

Renaissance in CNN use over the last few years, with increasingly complex network-in-network models that allow for deeper learning of more complex features.



The "GoogleNet" circa 2014

The brilliance of this inception module is that it uses kernels of several sizes but keeps the number of feature maps under control by use of a 1x1 convolution.

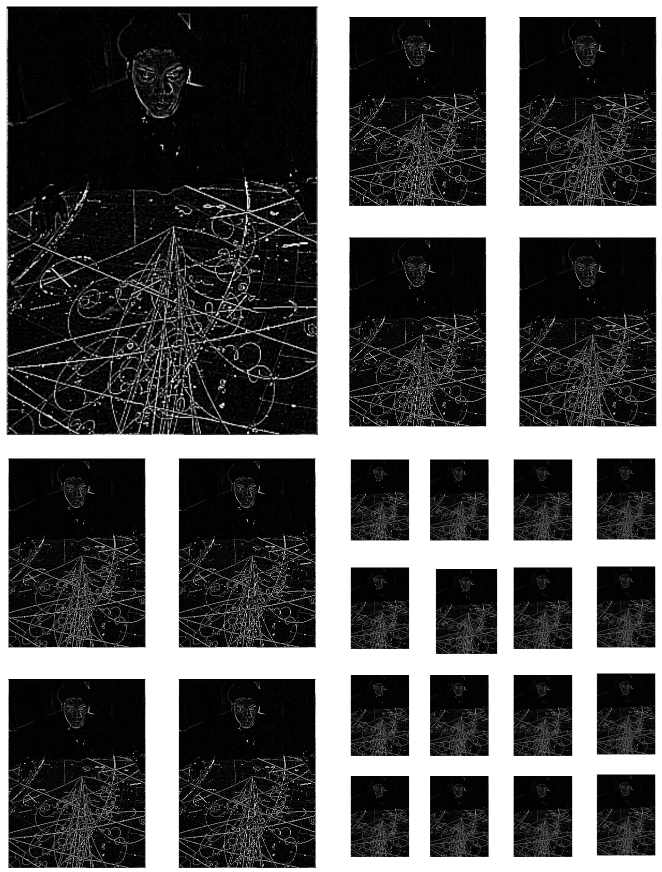
Superhuman Performance

Some examples from one of the early breakout CNNs.
Googles latest “Inception-v3” net achieves 3.46% top 5 error rate on the image net dataset. Human performance is at ~5%.

Deep Learning at NOvA

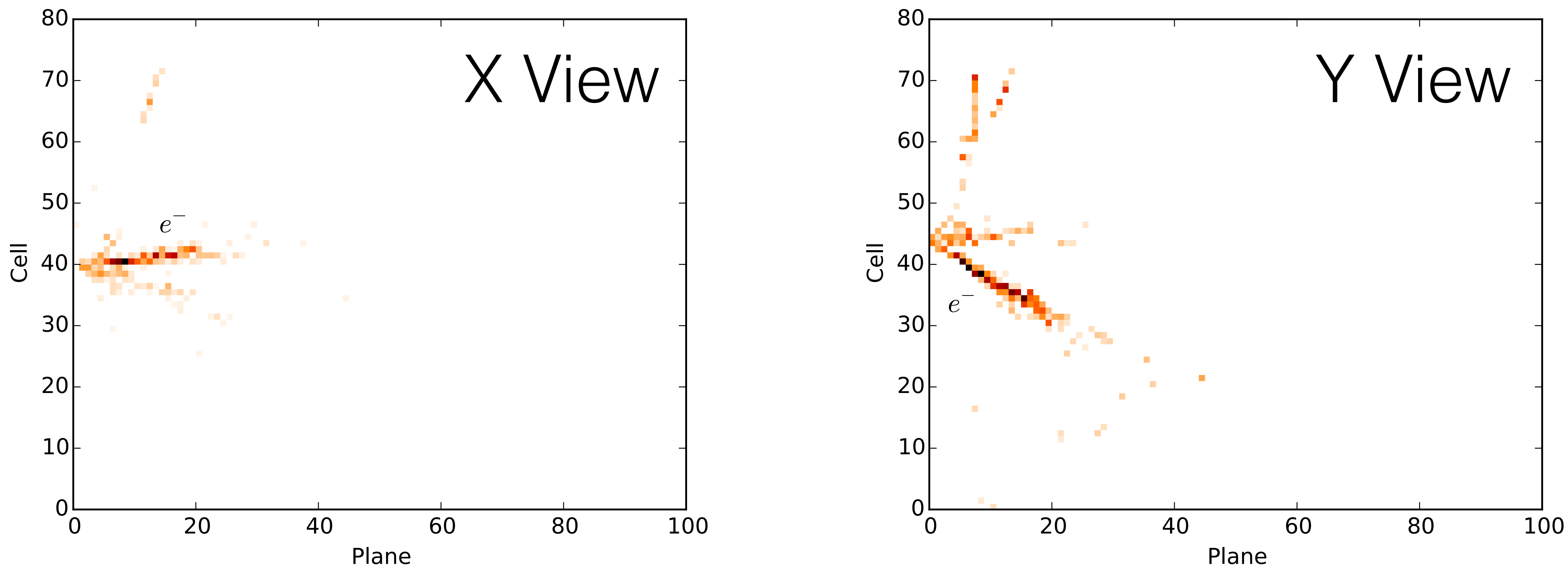
Deep Learning

What if we use the tools of the deep learning and computer vision communities to try and classify events?

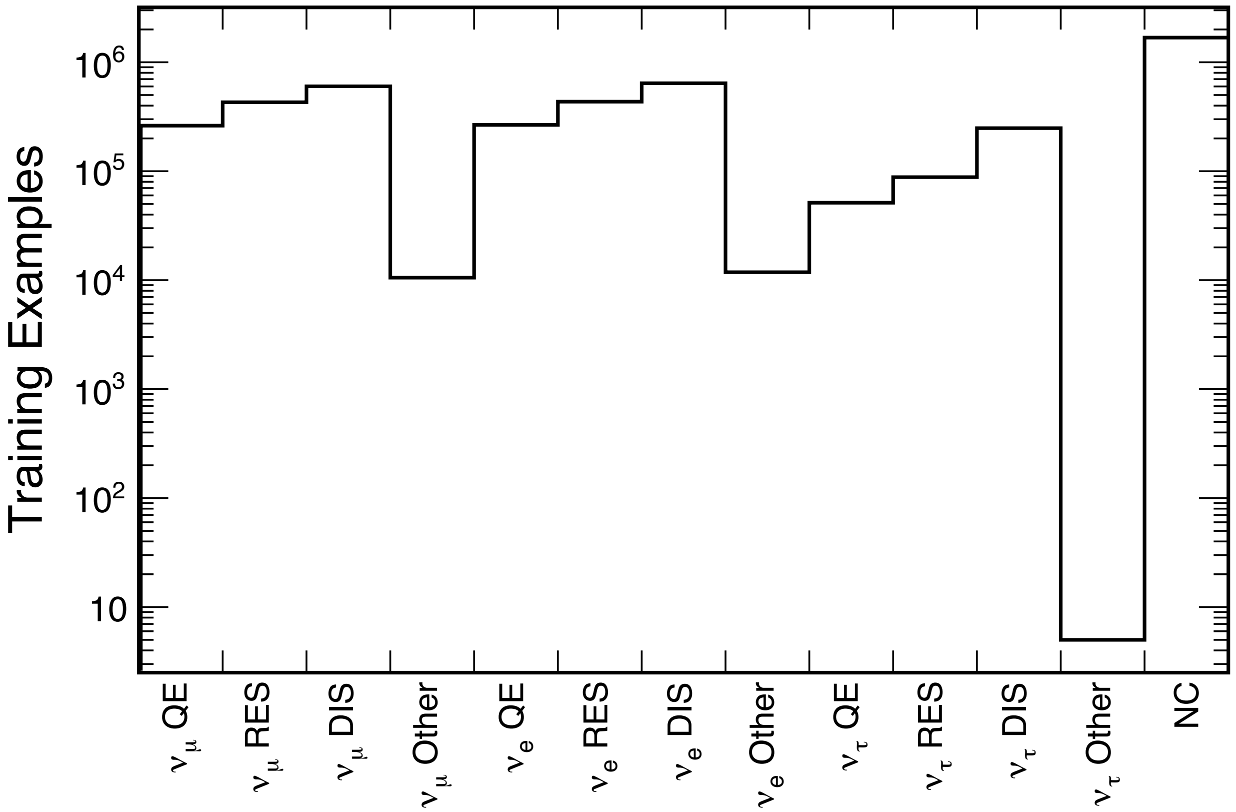


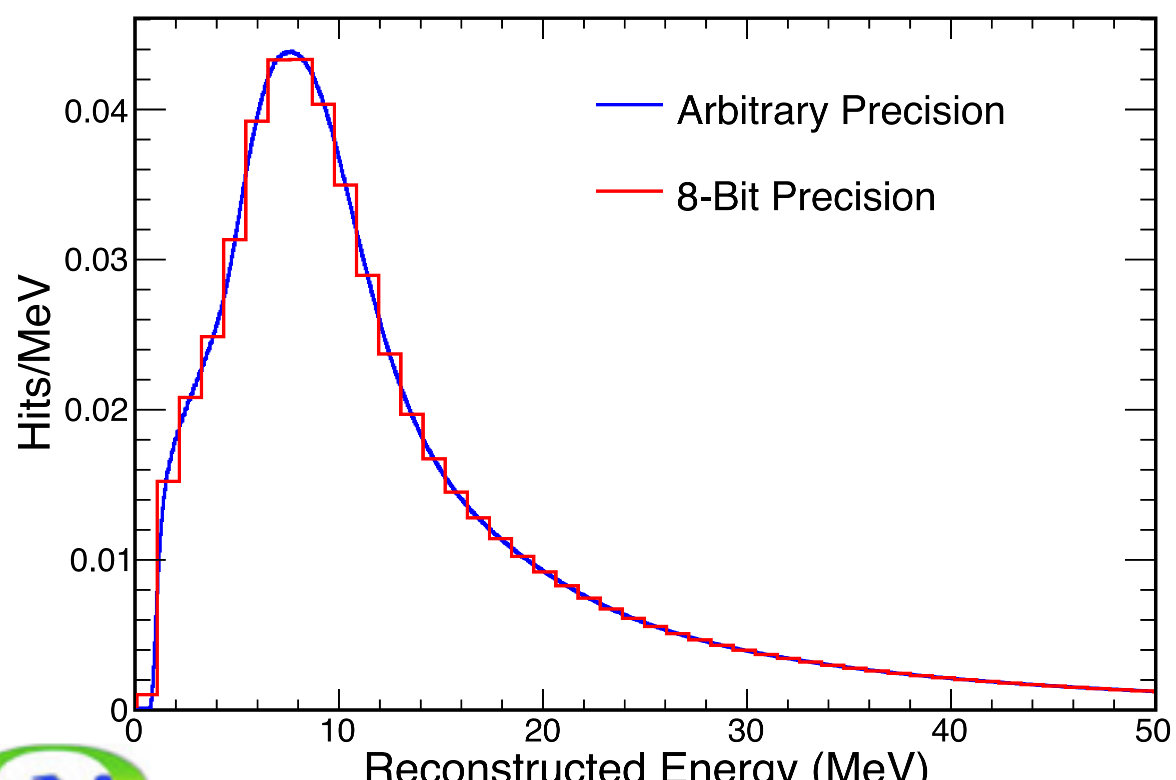
Our Input

Our input "image", is a pair of maps of the hits in a tight space/time window. One for the X view and another for the Y view. Each "pixel" is the calibrated energy response in that cell. All "images" have the same dimensions- 100 planes by 80 cells.



The Training Sample





- 4.7 million, minimally preselected simulated events, pushed into LevelDB databases: 80% for training and 20% for testing.
- Rescale calibrated energy depositions to go from 0 to 255 and truncate to chars for dramatically reduced file size at no loss of information
- Fine tuned with 5 million cosmic data events taken from an out of beam time minimal bias trigger.

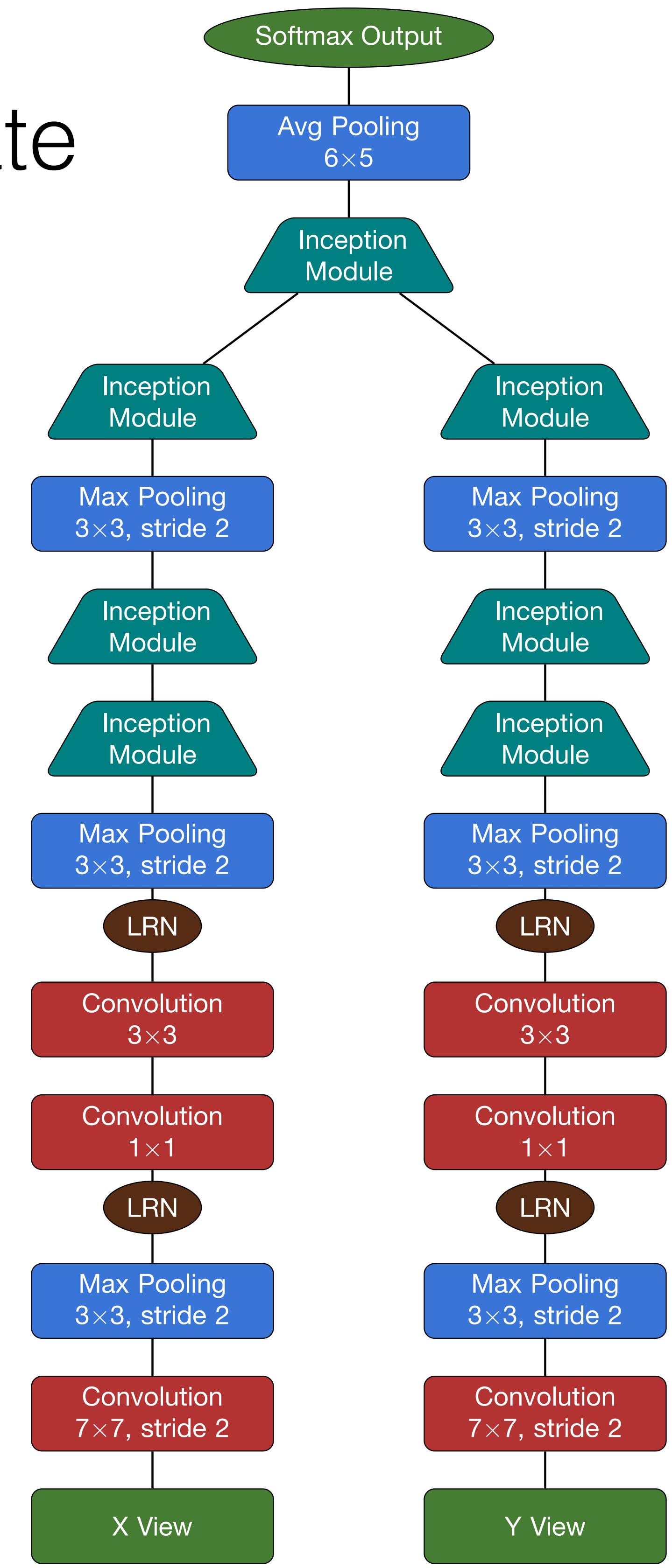
Our Architecture

Based on the first googlenet. Largest innovation is splitting each view into a separate sequence of layers and concatenating the outputs near the end of the network. Named “Convolutional Visual Network”, or **CVN**.

The architecture attempts to categorize events as $\{\nu_\mu, \nu_e, \nu_\tau\} \times \{QE, RES, DIS\}$, NC, or Cosmogenic.

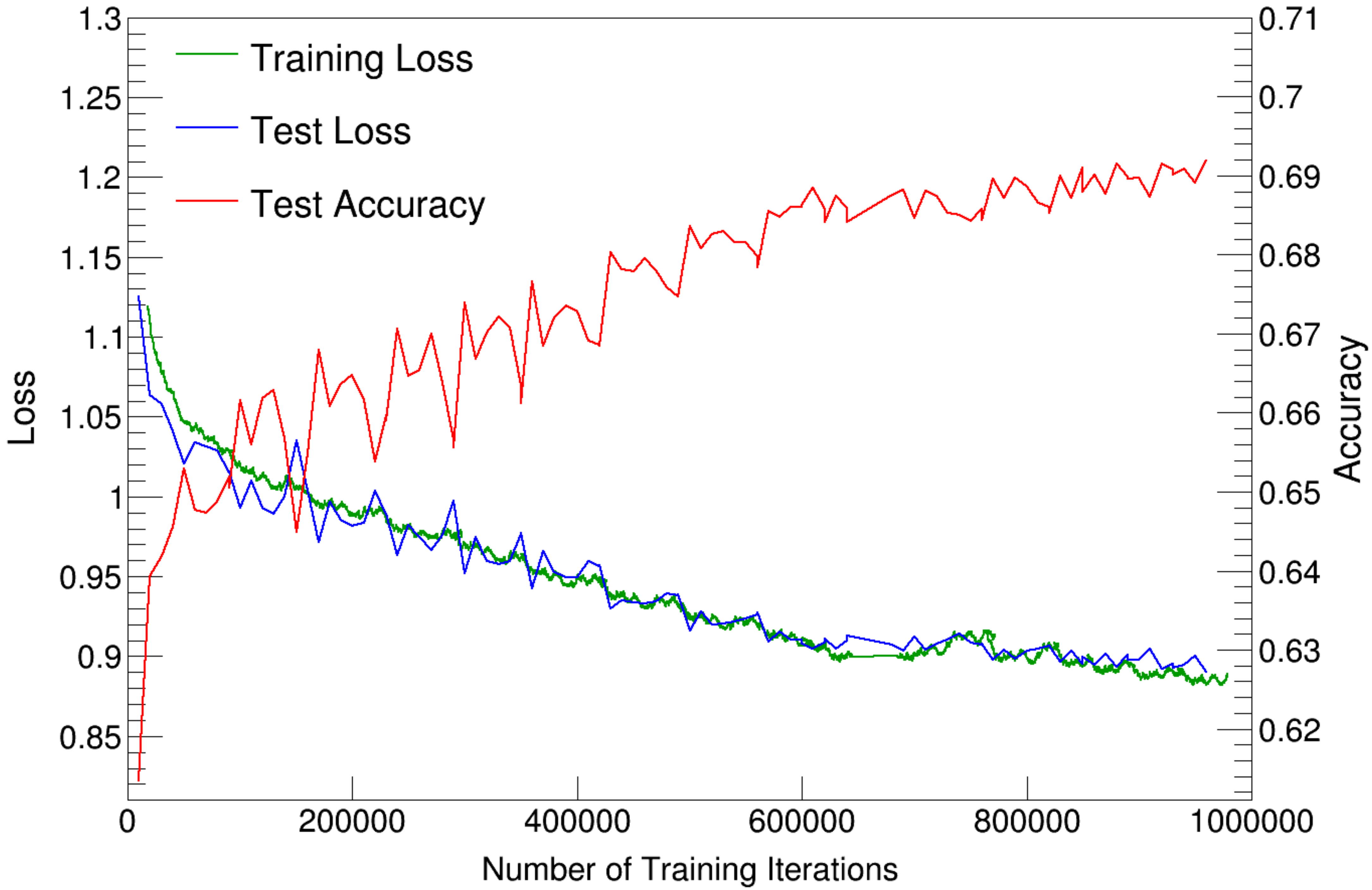
Designed to be a universal classifier.

Built in the excellent CAFFE framework: http://caffe.berkeleyvision.org/

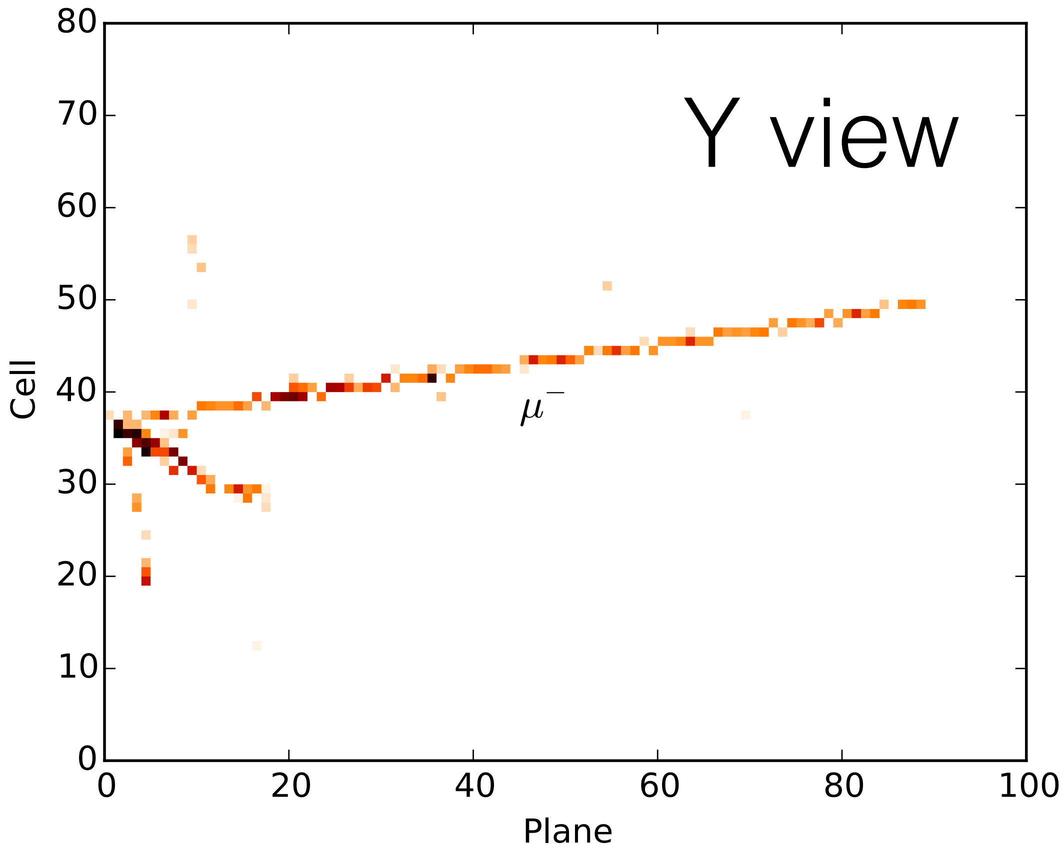


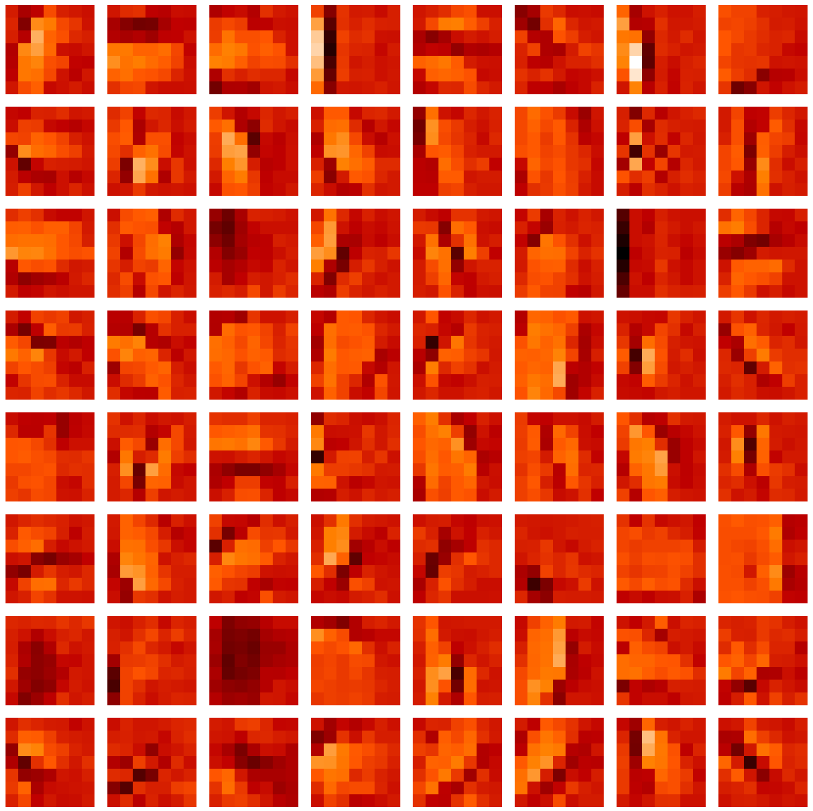
Training Performance

No sign of overtraining- exceptional training test set performance agreement!



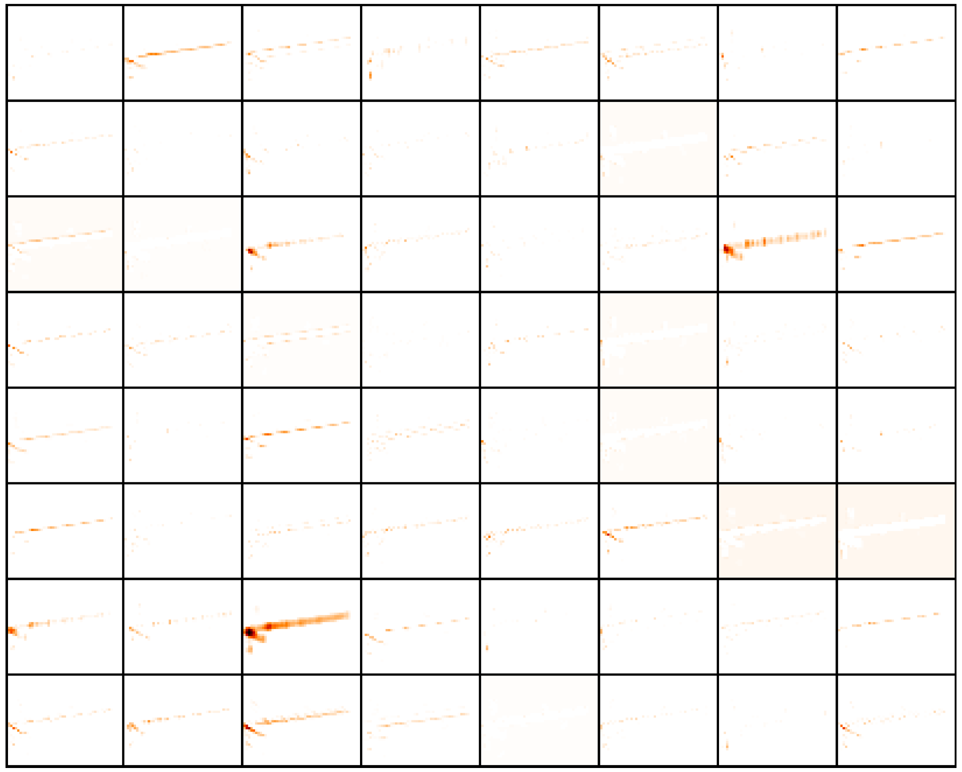
Example CVN Kernels In Action: First Convolution





X

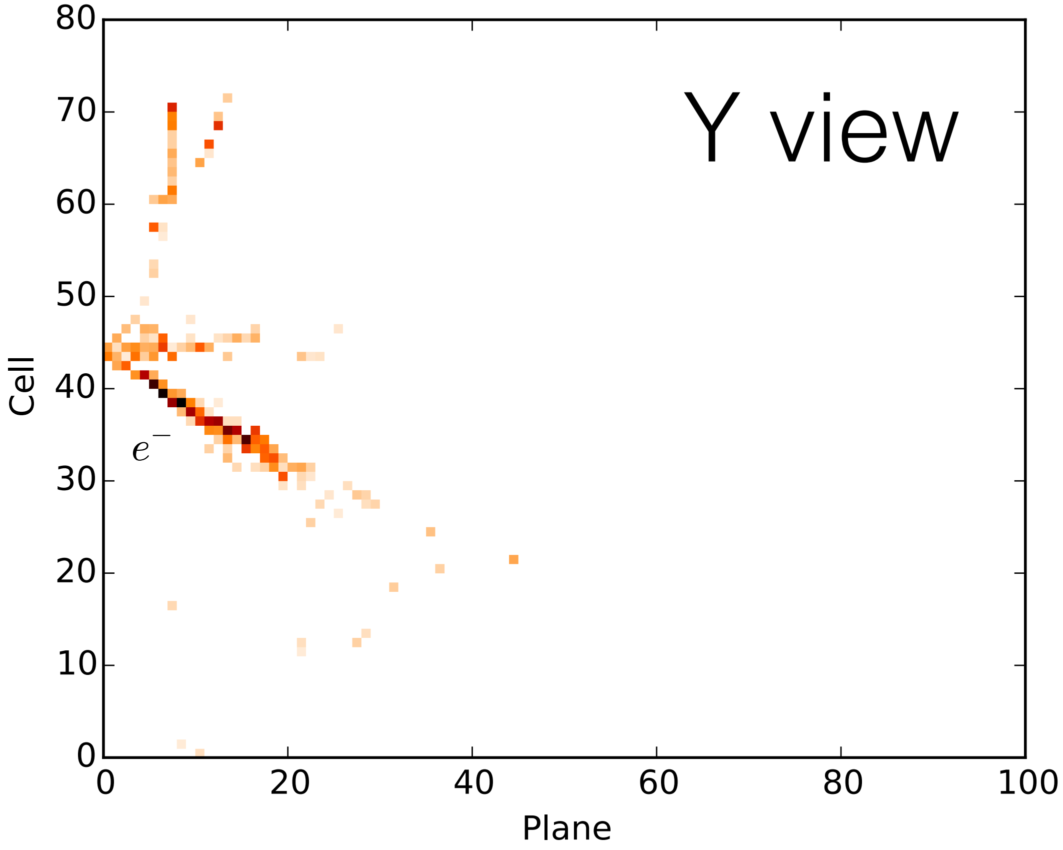
=



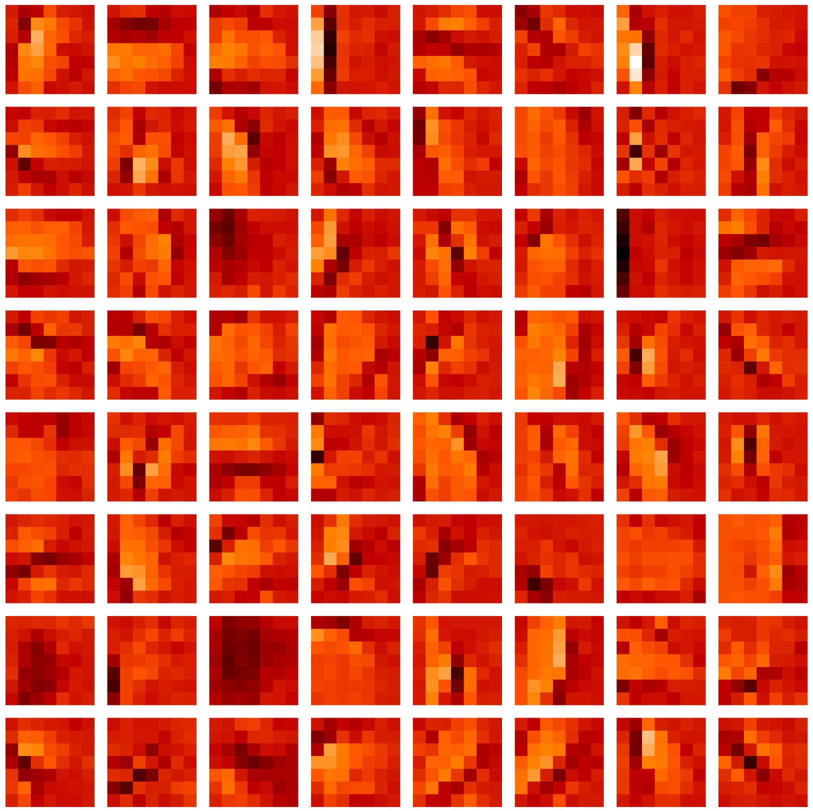
Here the earliest convolutional layer in the network starts by pulling out primitive shapes and lines.

Already “showers” and “tracks” are starting to form.

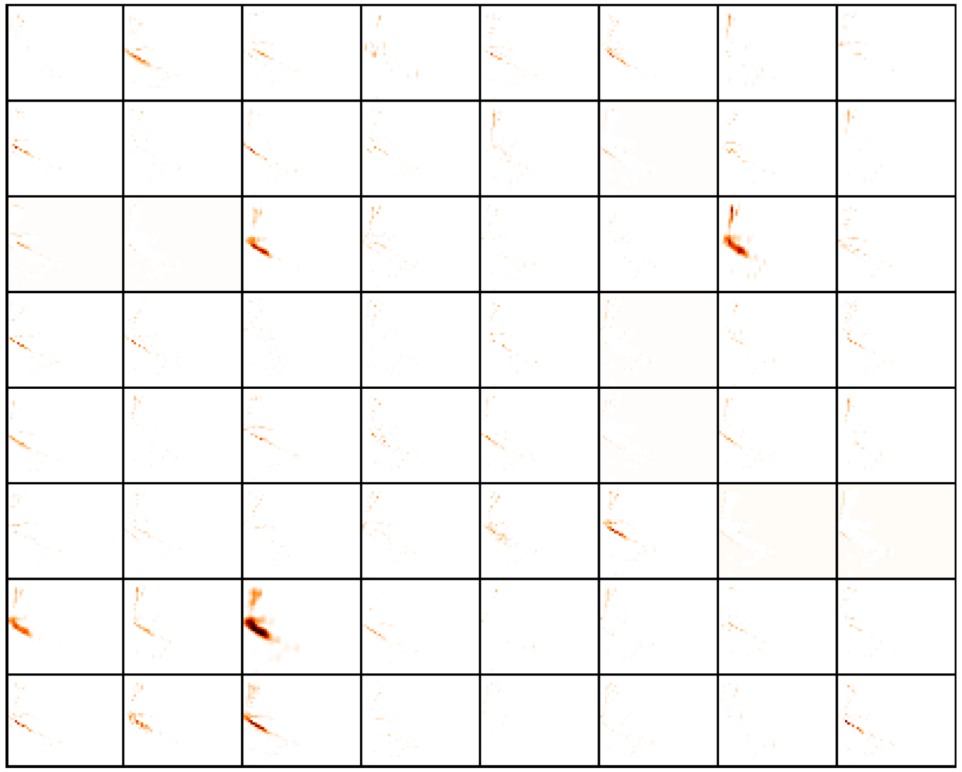
Example CVN Kernels In Action: First Convolution



X



=



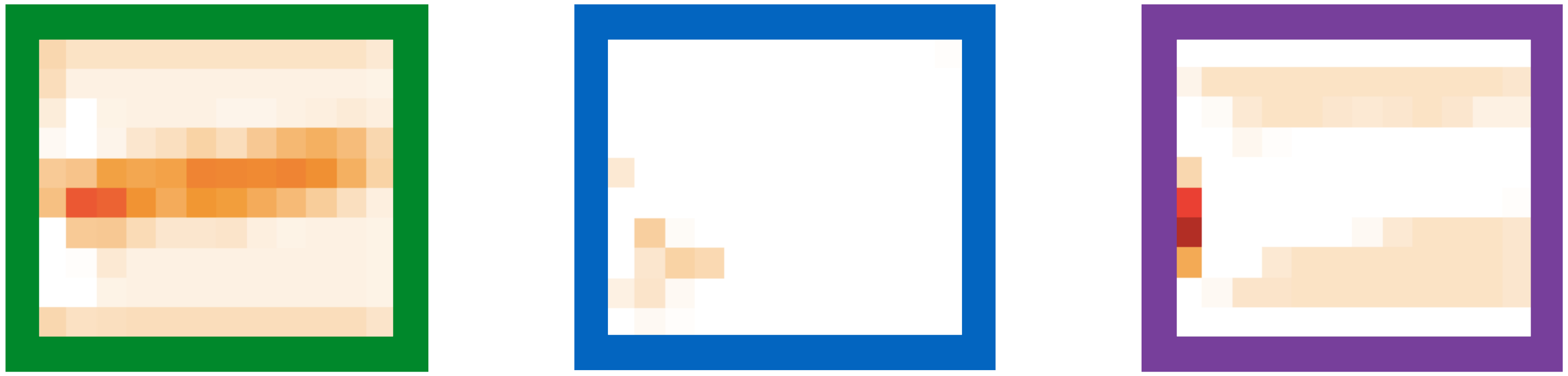
Here the earliest convolutional layer in the network starts by pulling out primitive shapes and lines.

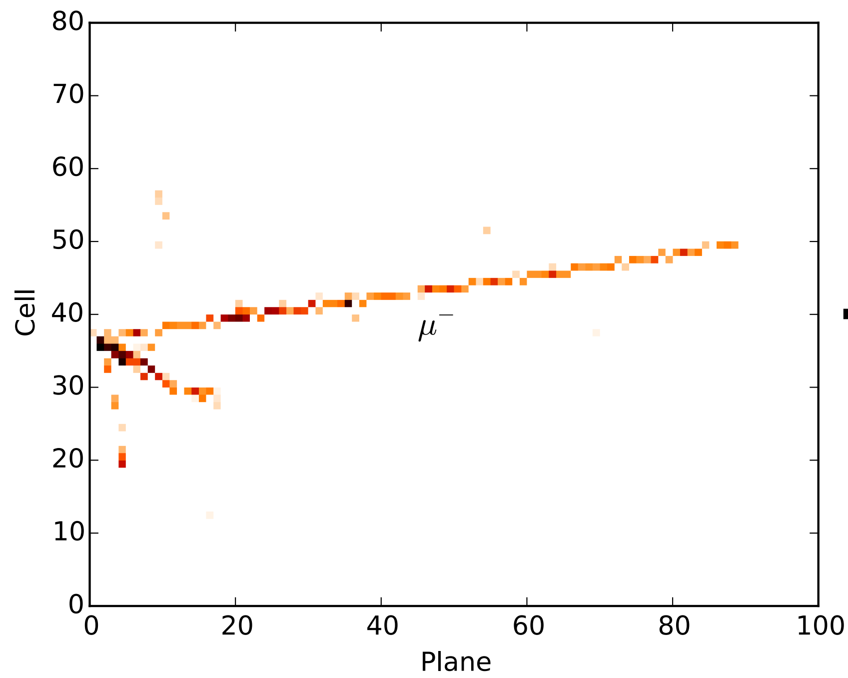
Already "showers" and "tracks" are starting to form.

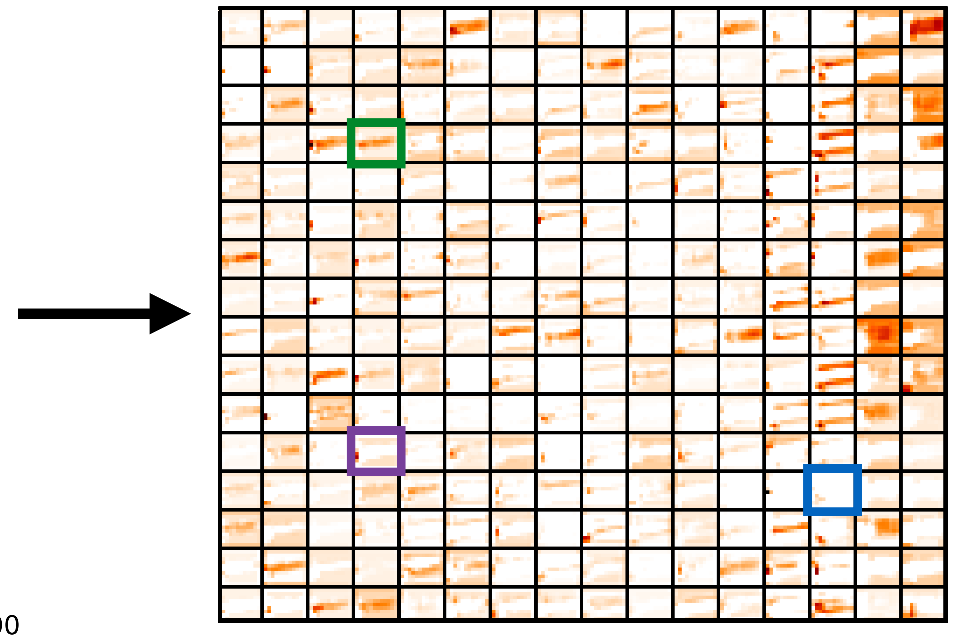
Example CVN Kernels In Action: First Inception Module Output

Deeper in the network, now after the first inception module we can see more complex features have started to be extracted.

Some seem particularly sensitive to muon tracks, EM showers, or hadronic activity.



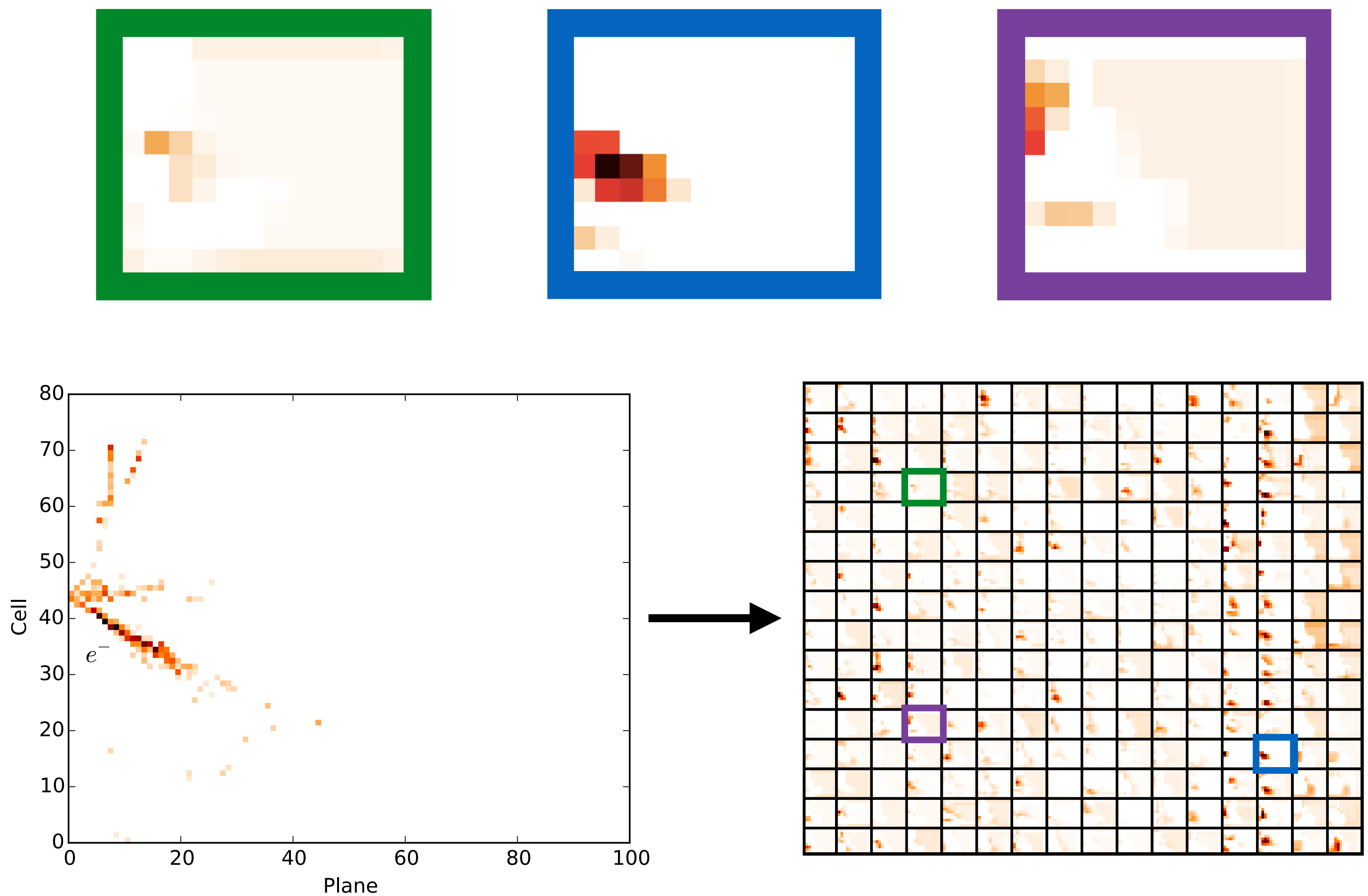




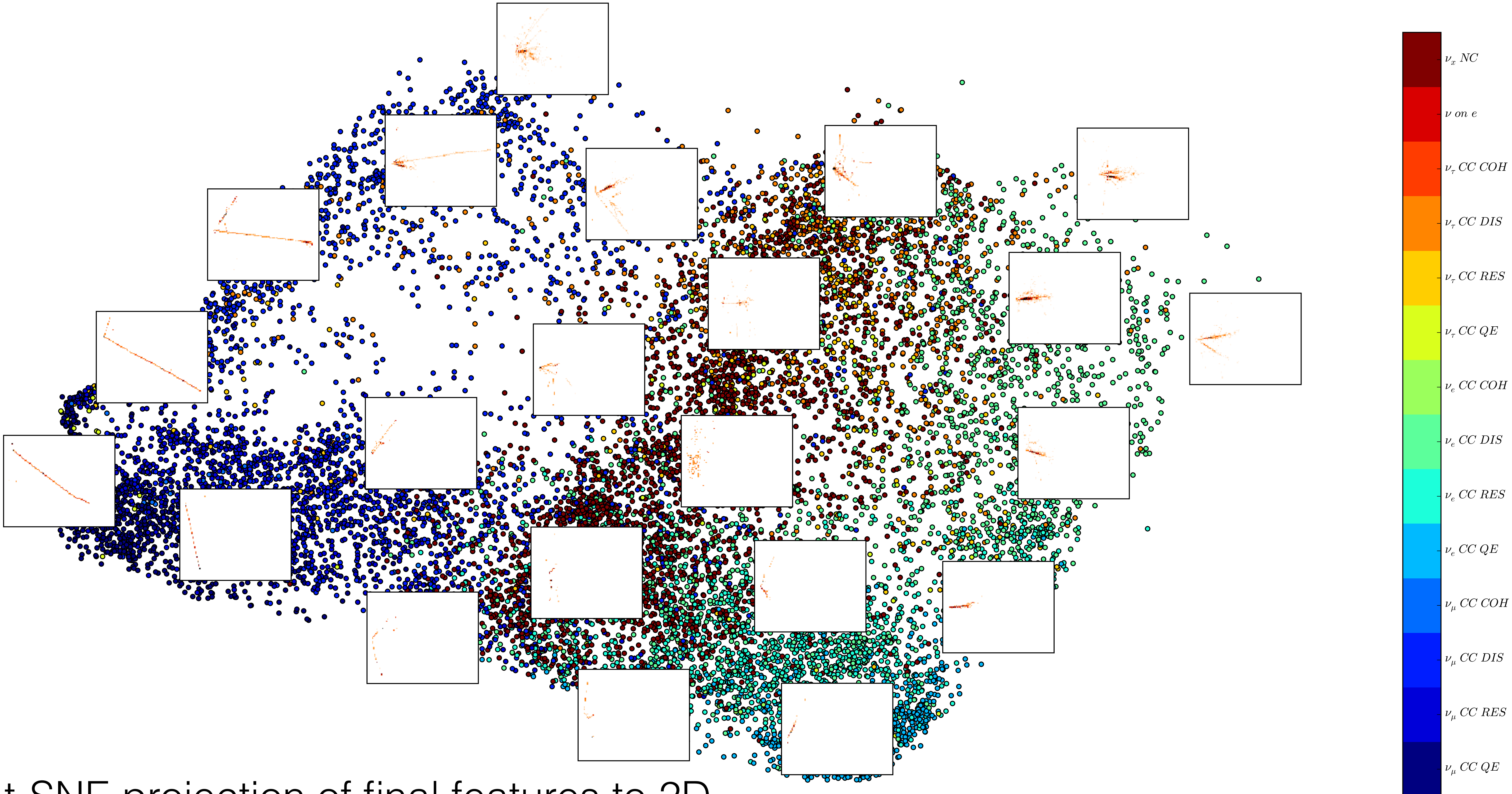
Example CVN Kernels In Action: First Inception Module Output

Deeper in the network, now after the first inception module we can see more complex features have started to be extracted.

Some seem particularly sensitive to muon tracks, EM showers, or hadronic activity.

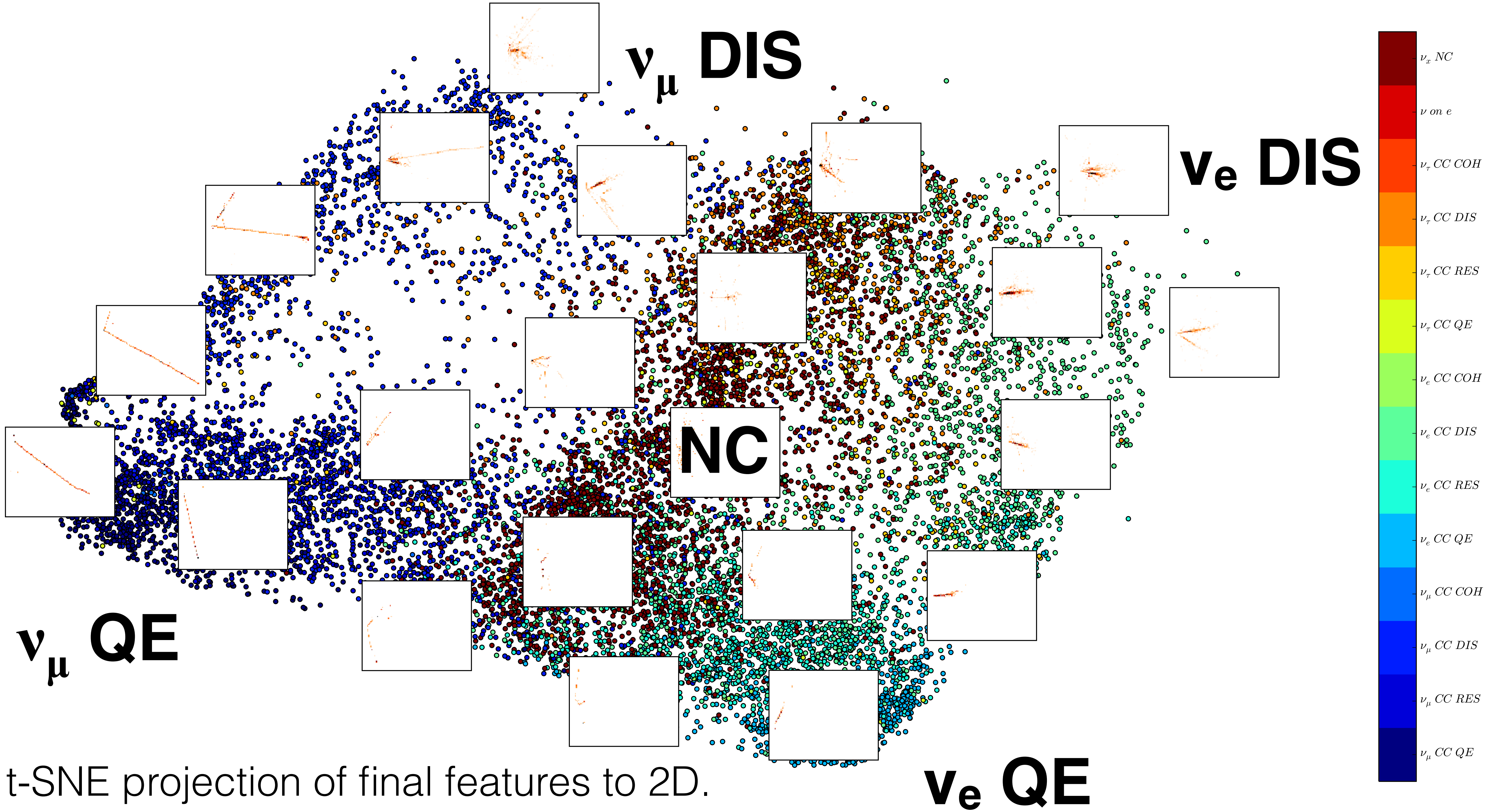


t-SNE Representation of Test Sample



t-SNE projection of final features to 2D.
Truth labels, training sample subset.

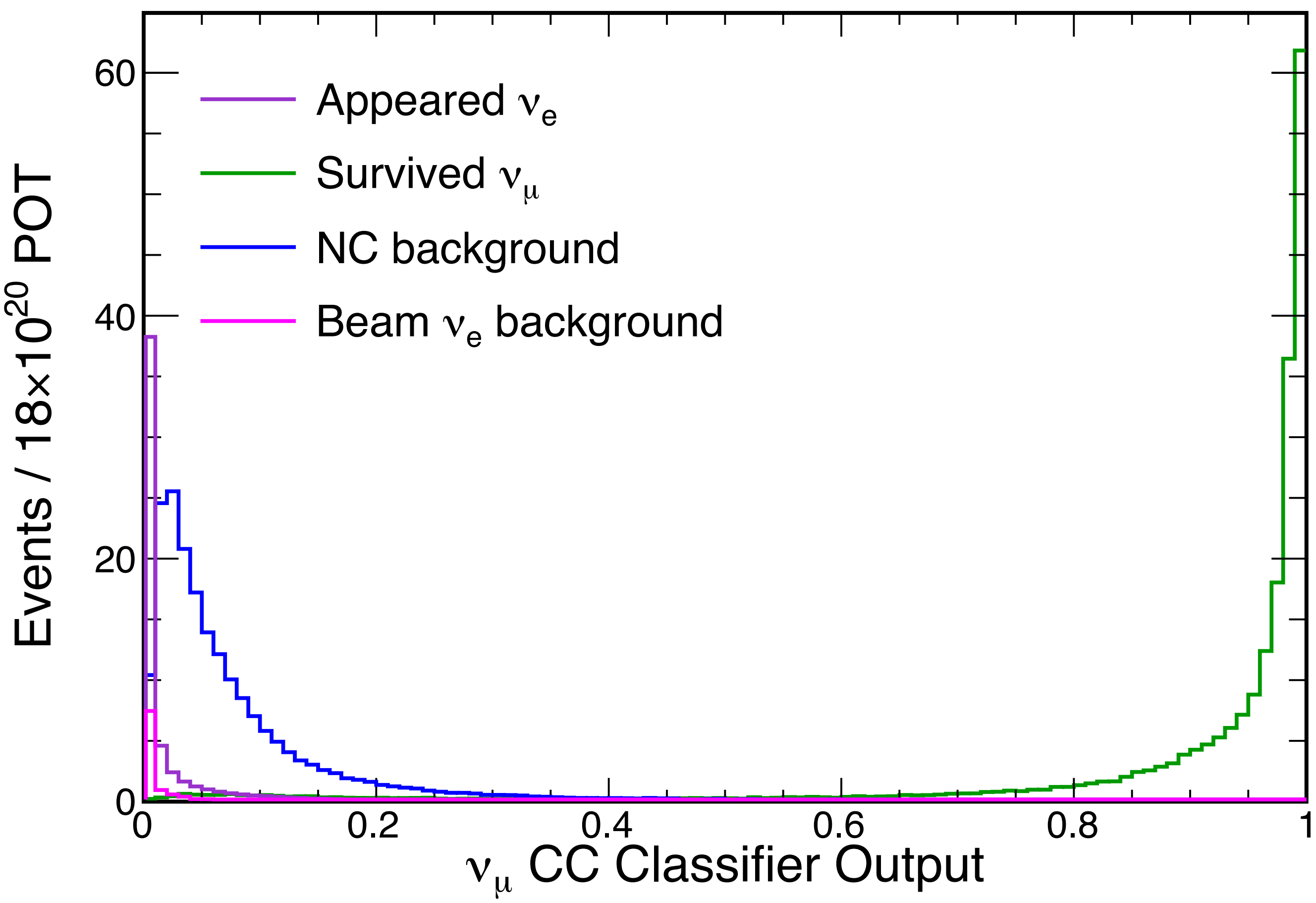
t-SNE Representation of Test Sample

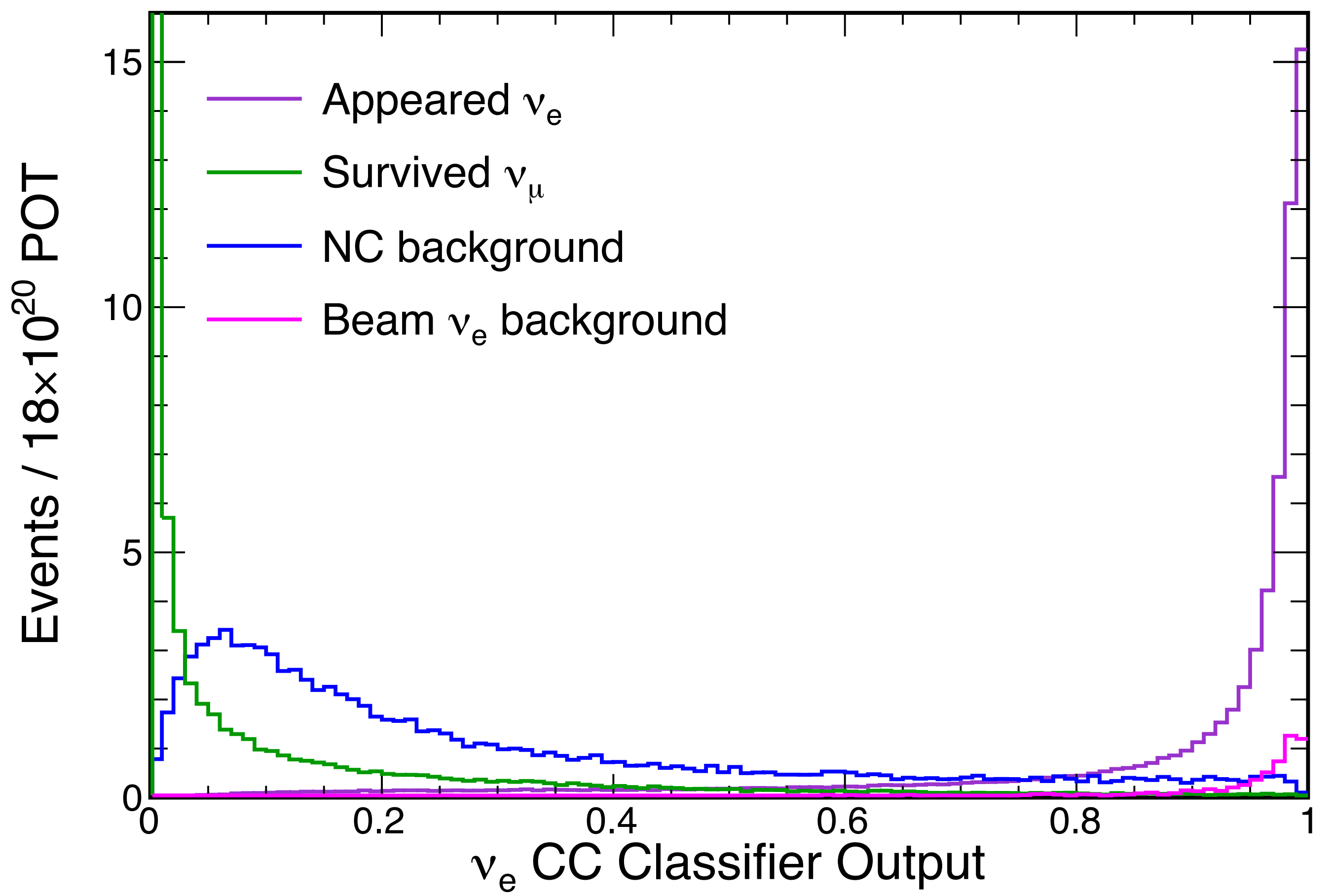


t-SNE projection of final features to 2D.
Truth labels, training sample subset.

The Bottom Line

After oscillations, cosmic rejection cuts, data quality cuts:

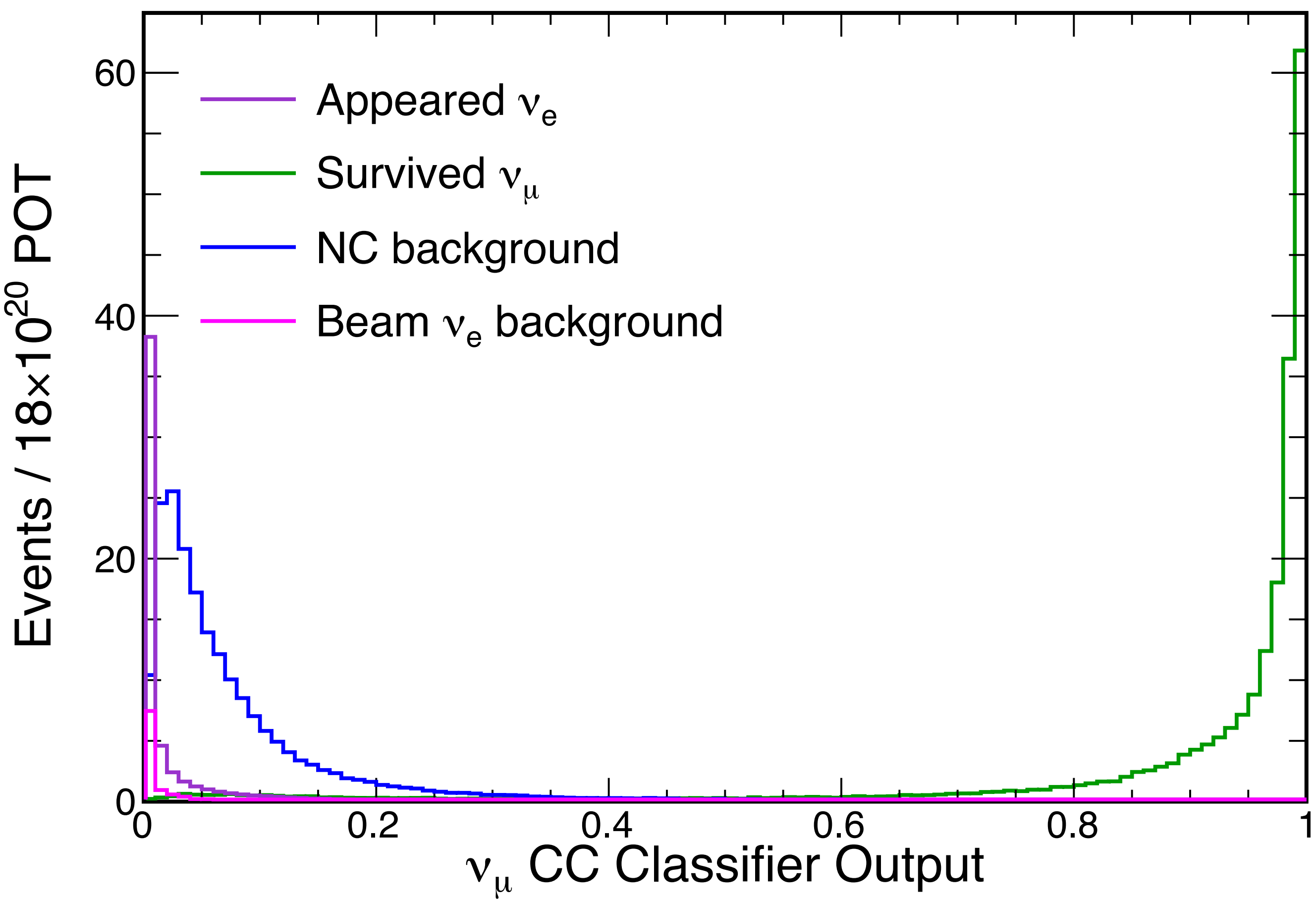


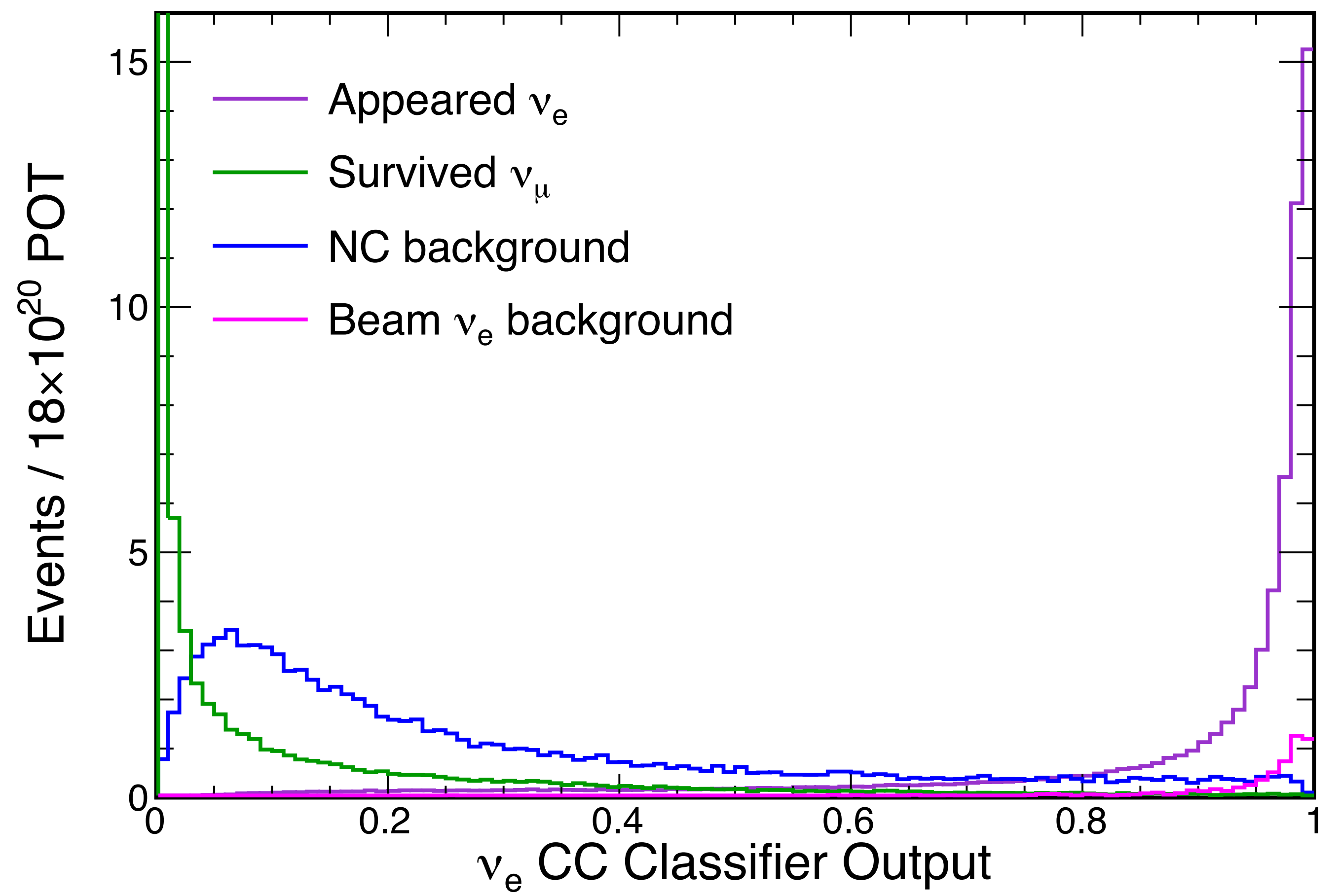


Excellent separation of the $\boldsymbol{\nu_\mu}$ sample, but ~identical to existing, much simpler, KNN selector. Matches expectation- hard to miss a muon track. Space to improve in cosmic rejection.

The Bottom Line

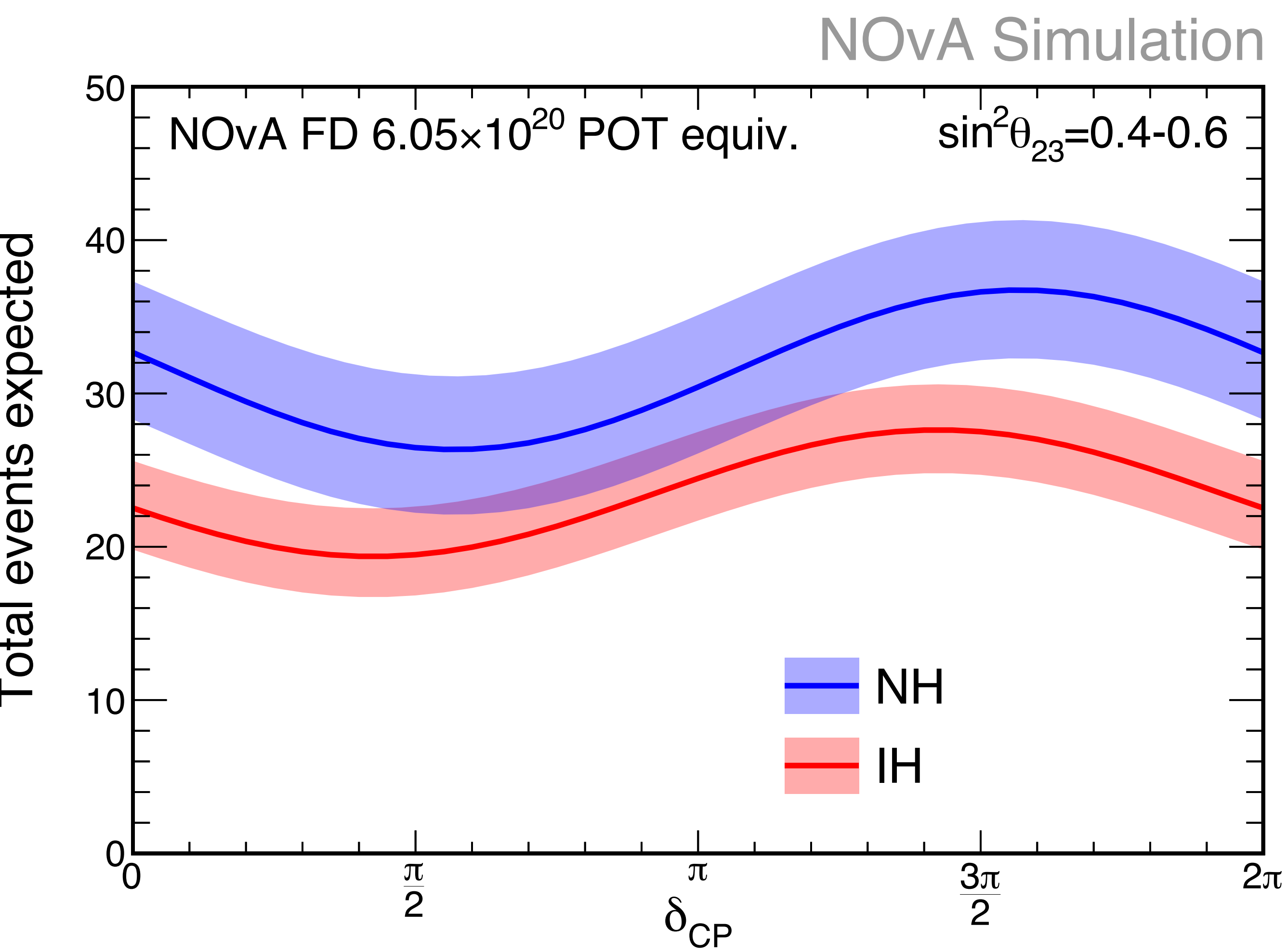
After oscillations, cosmic rejection cuts, data quality cuts:

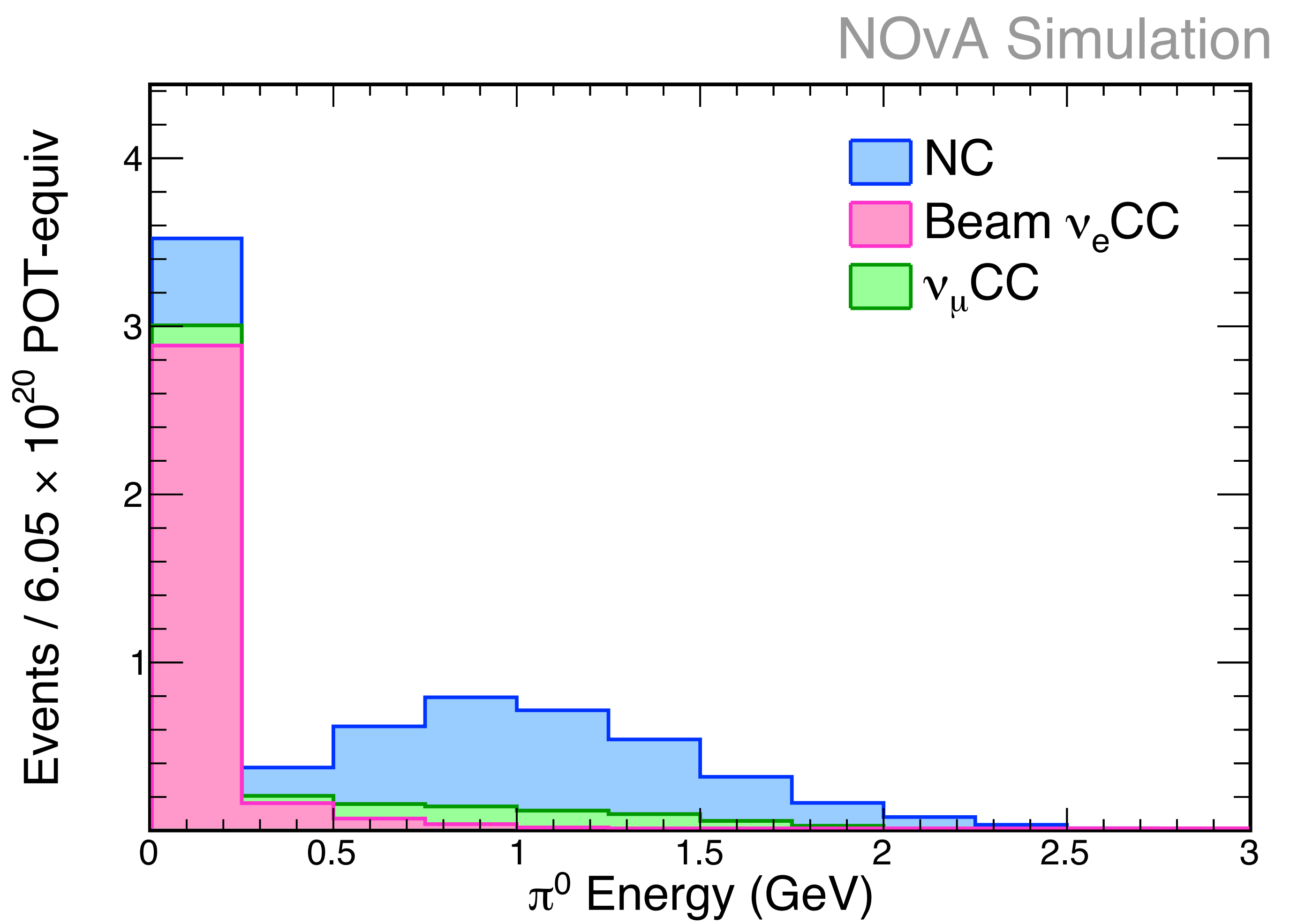




However our CNN achieves **73%** efficiency and **76%** purity on **ν_e** selection at the $s/\sqrt{s+b}$ optimized cut.
Equivalent to **30%** more exposure with the old PIDs.

ν_e Selection Performance



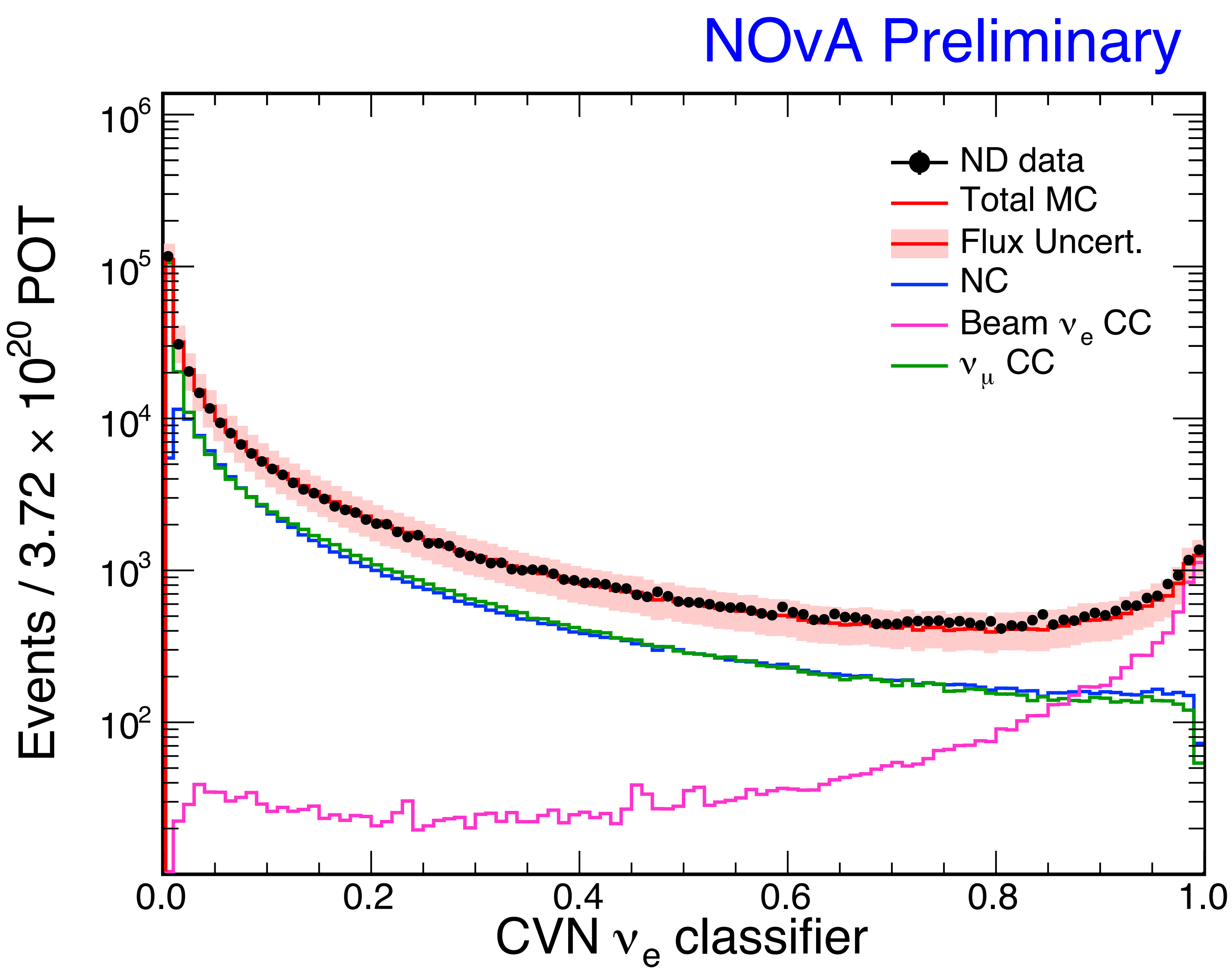


Total bkg	NC	Beam ν_e	ν_μCC	ν_τCC	Cosmogenic
8.2	3.7	3.1	0.7	0.1	0.5

Greater than 90% of beam backgrounds contain an electromagnetic shower.

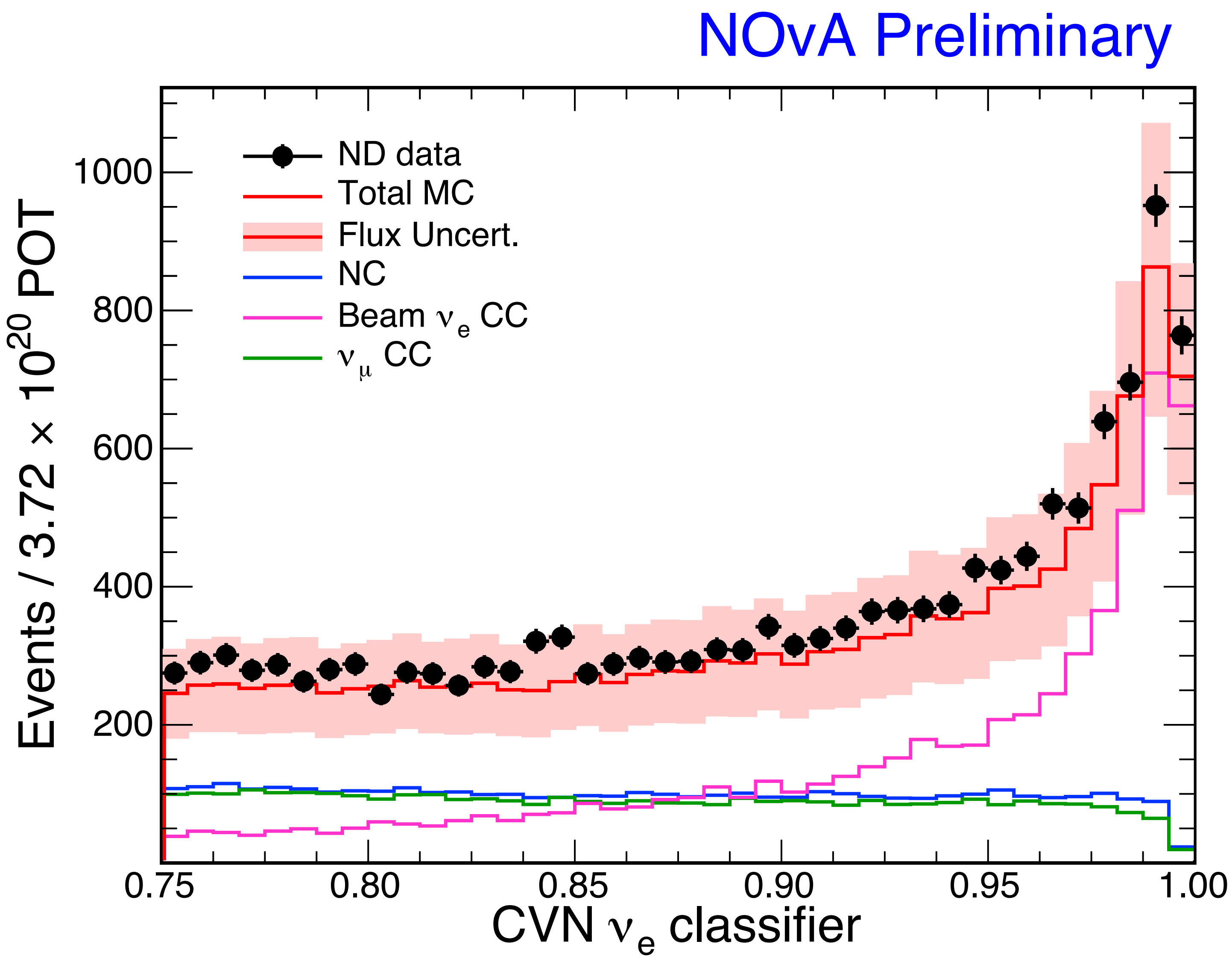
CVN in Use For the ν_e Appearance Analysis

Data Driven Cross Checks: The Near Detector



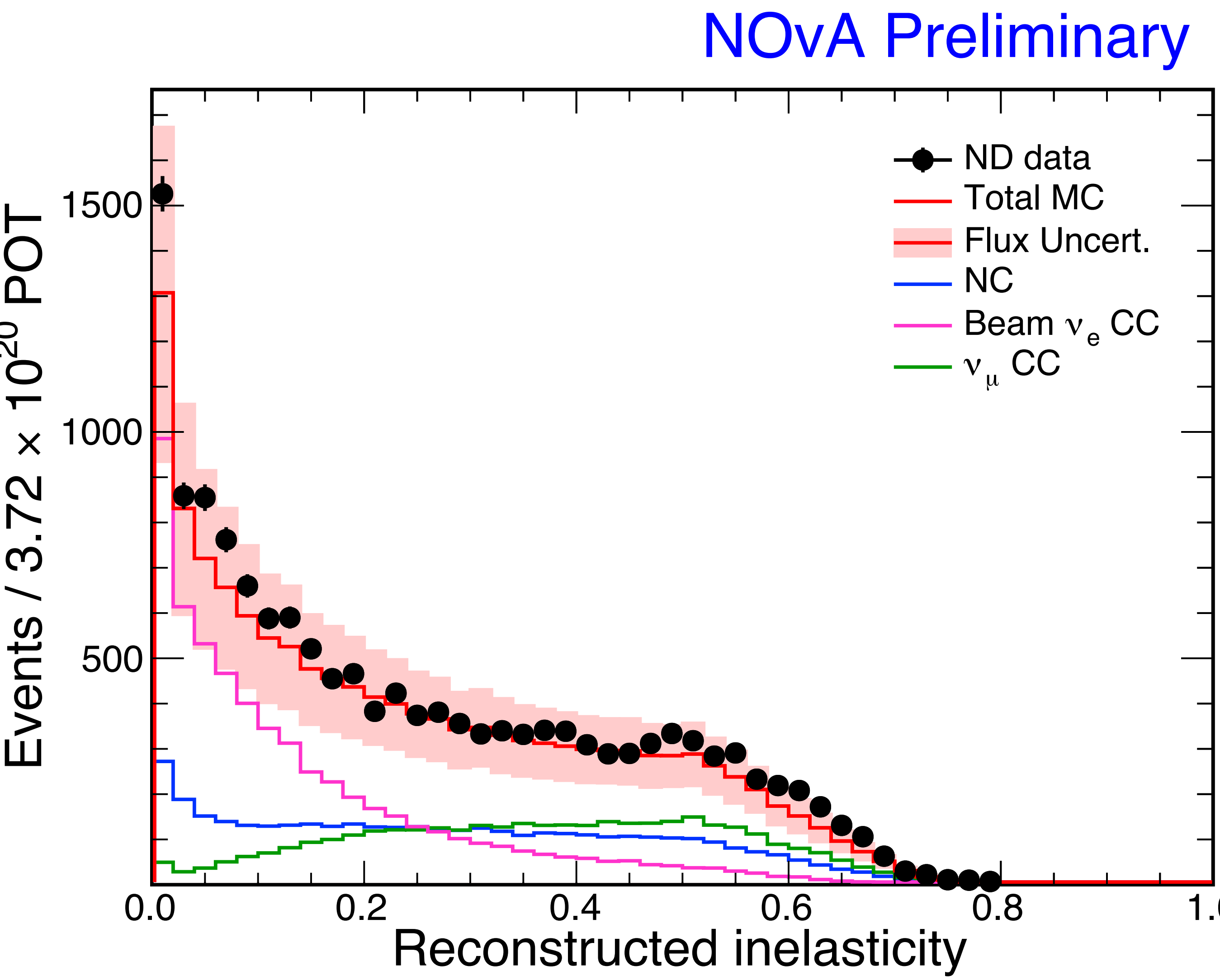
Our most powerful tool for understanding our background simulation is our functionally equivalent Near Detector. Excellent data/MC agreement across the board!

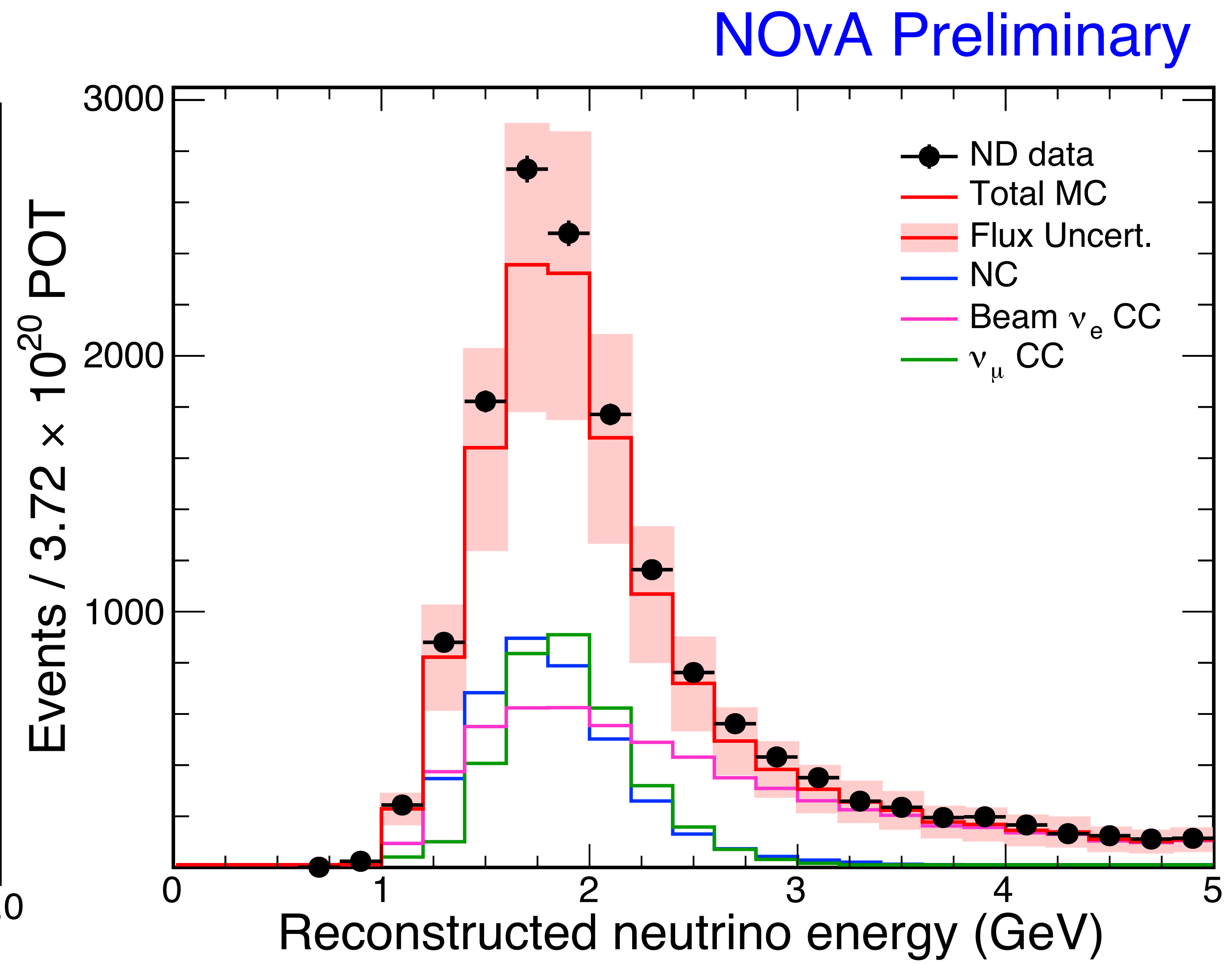
Data Driven Cross Checks: The Near Detector



Our most powerful tool for understanding our background simulation is our functionally equivalent Near Detector. Excellent data/MC agreement across the board!

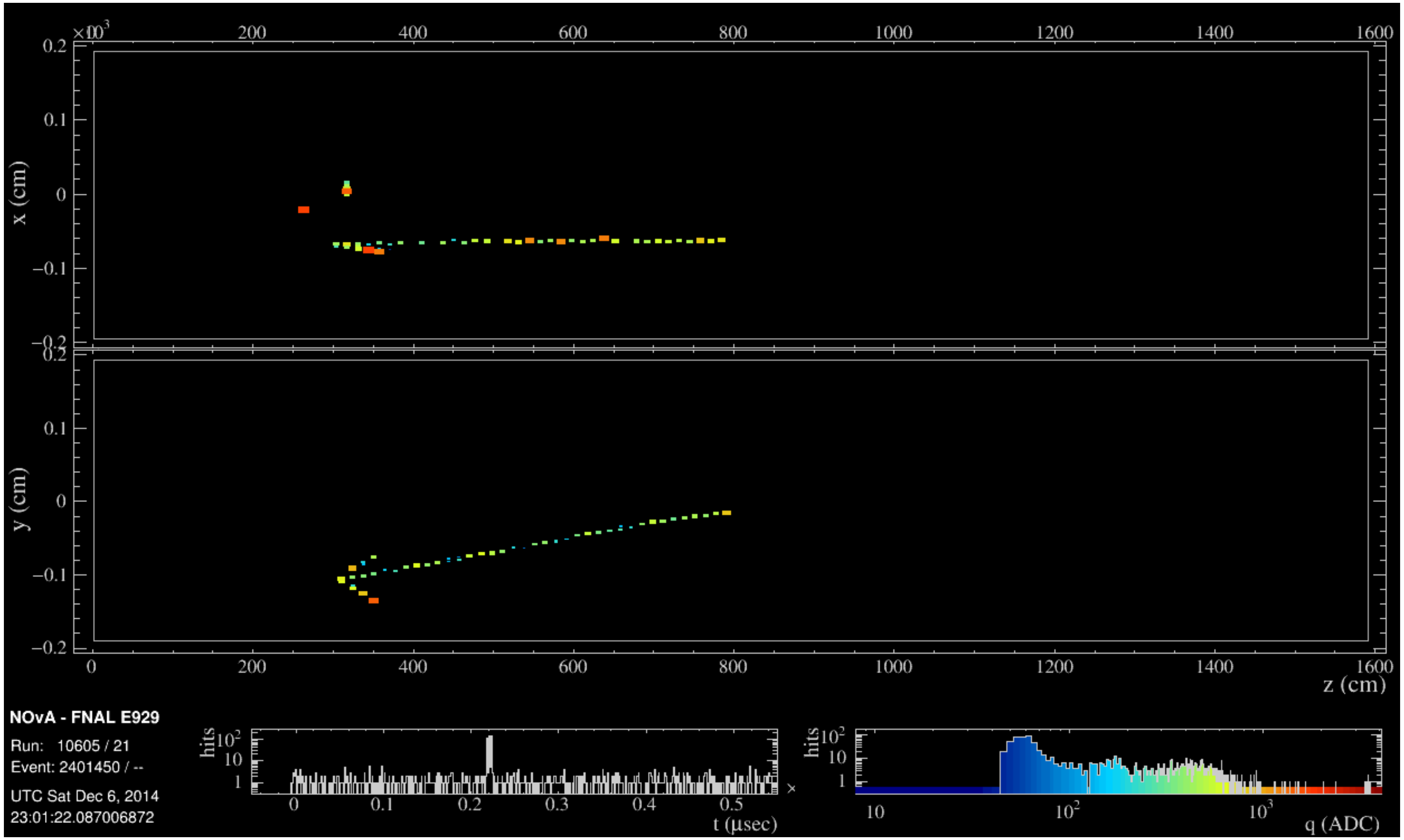
Data Driven Cross Checks: The Near Detector





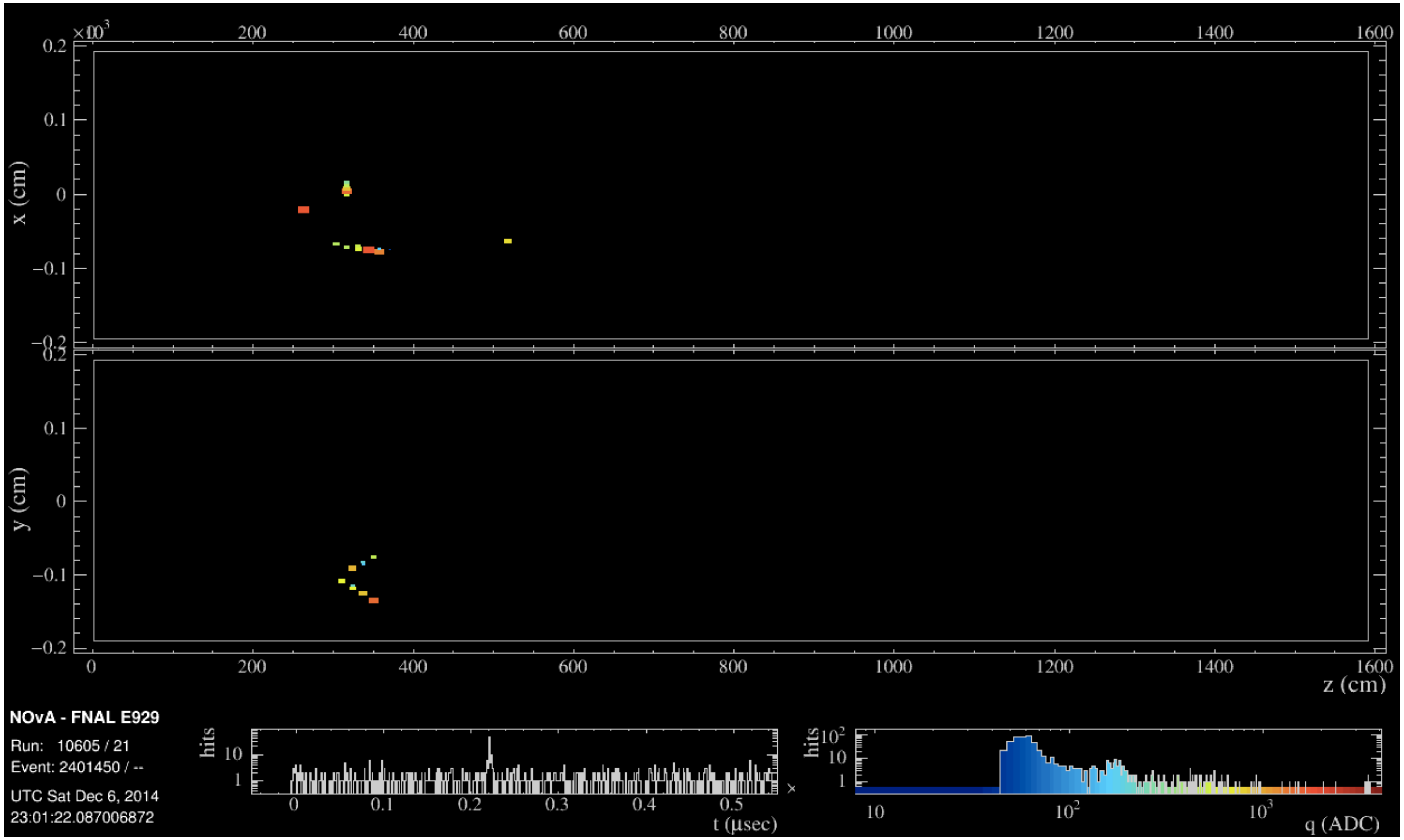
Our most powerful tool for understanding our background simulation is our functionally equivalent Near Detector. Excellent data/MC agreement across the board!

Data Driven Cross Checks: Muon Removed Electron Added



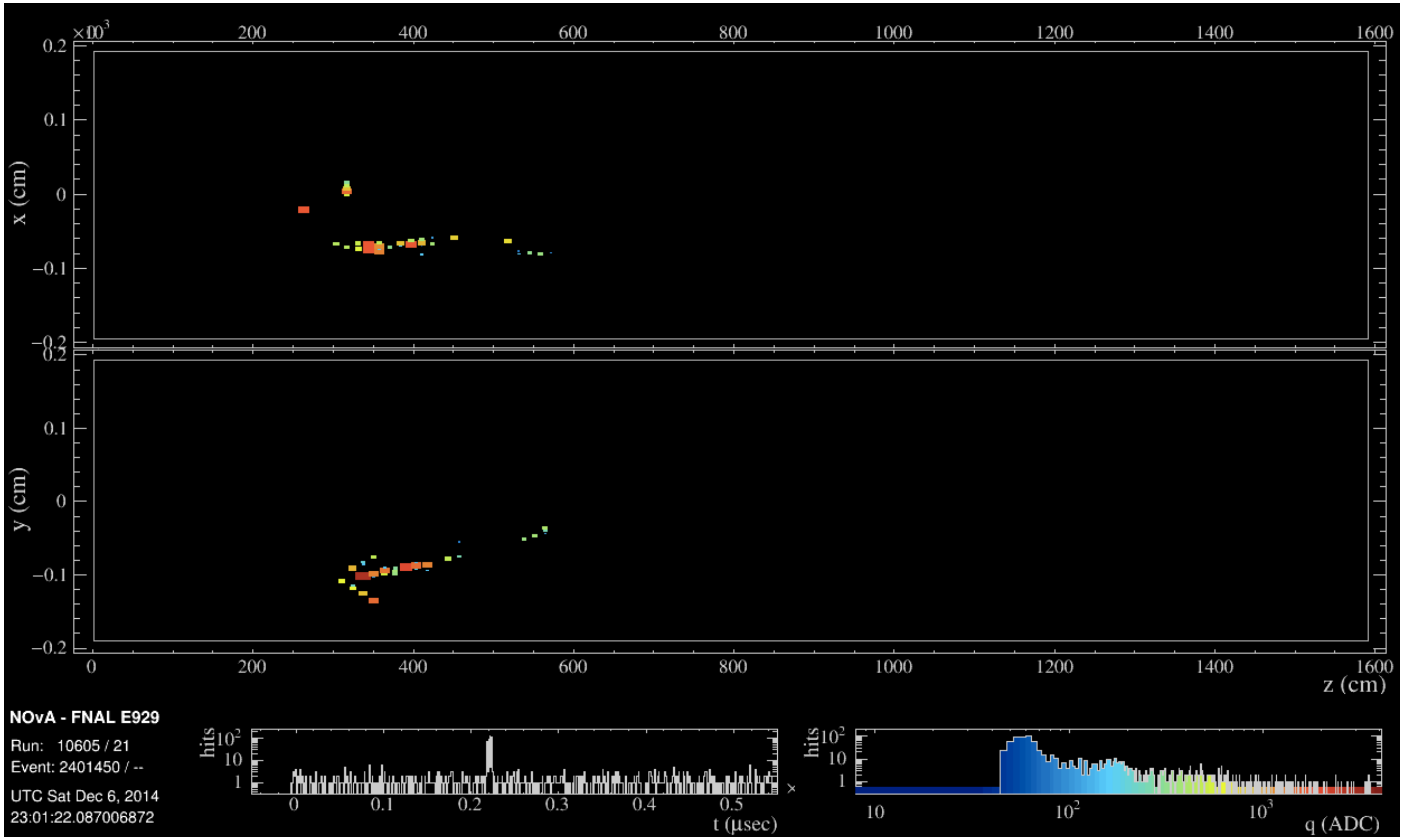
How to check our performance on our signal sample using the Near Detector? Try faking appeared electron neutrinos by creating hybrid data/simulation events.

Data Driven Cross Checks: Muon Removed Electron Added



How to check our performance on our signal sample using the Near Detector? Try faking appeared electron neutrinos by creating hybrid data/simulation events.

Data Driven Cross Checks: Muon Removed Electron Added



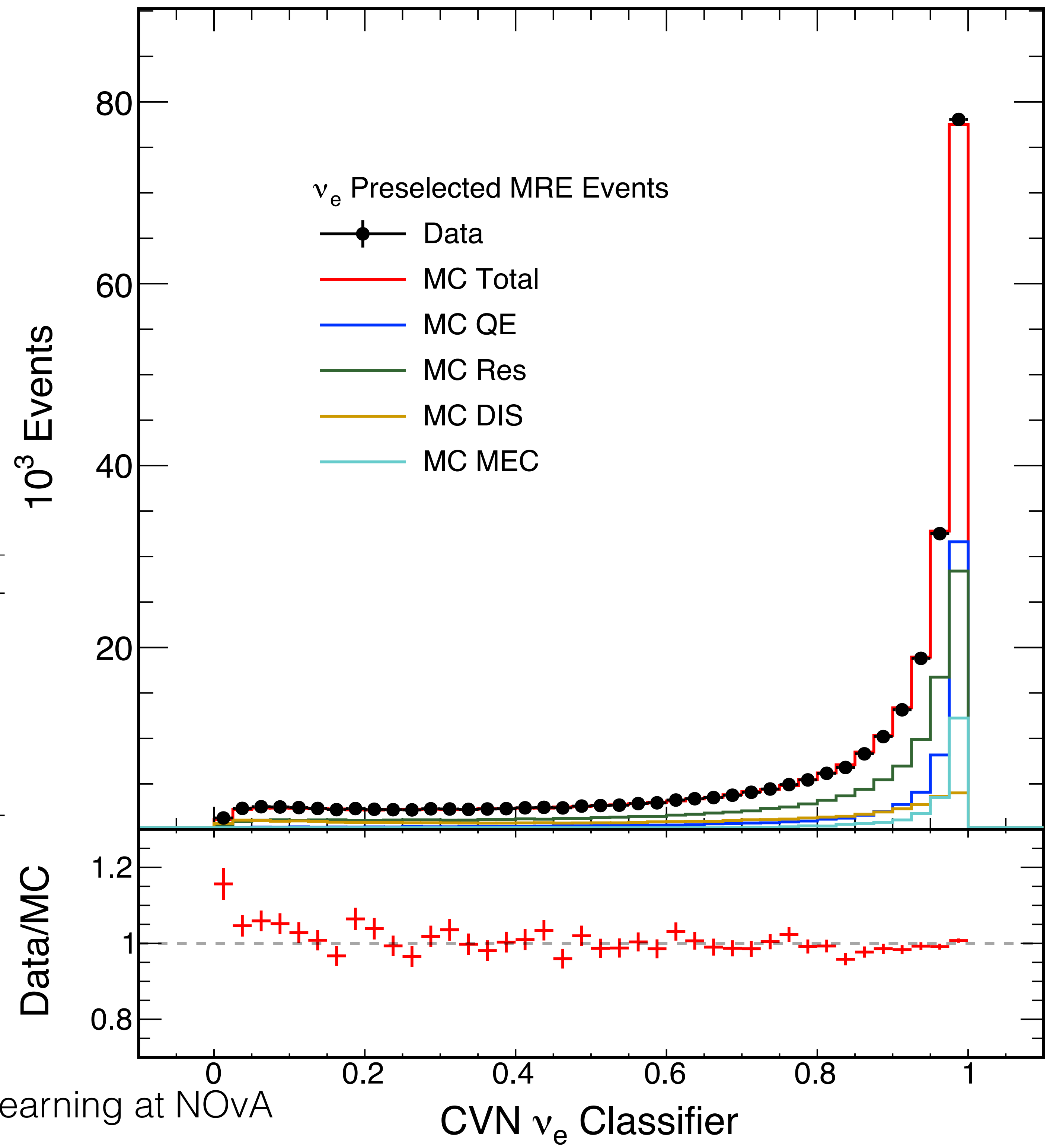
How to check our performance on our signal sample using the Near Detector? Try faking appeared electron neutrinos by creating hybrid data/simulation events.

Data Driven Cross Checks: Muon Removed Electron Added

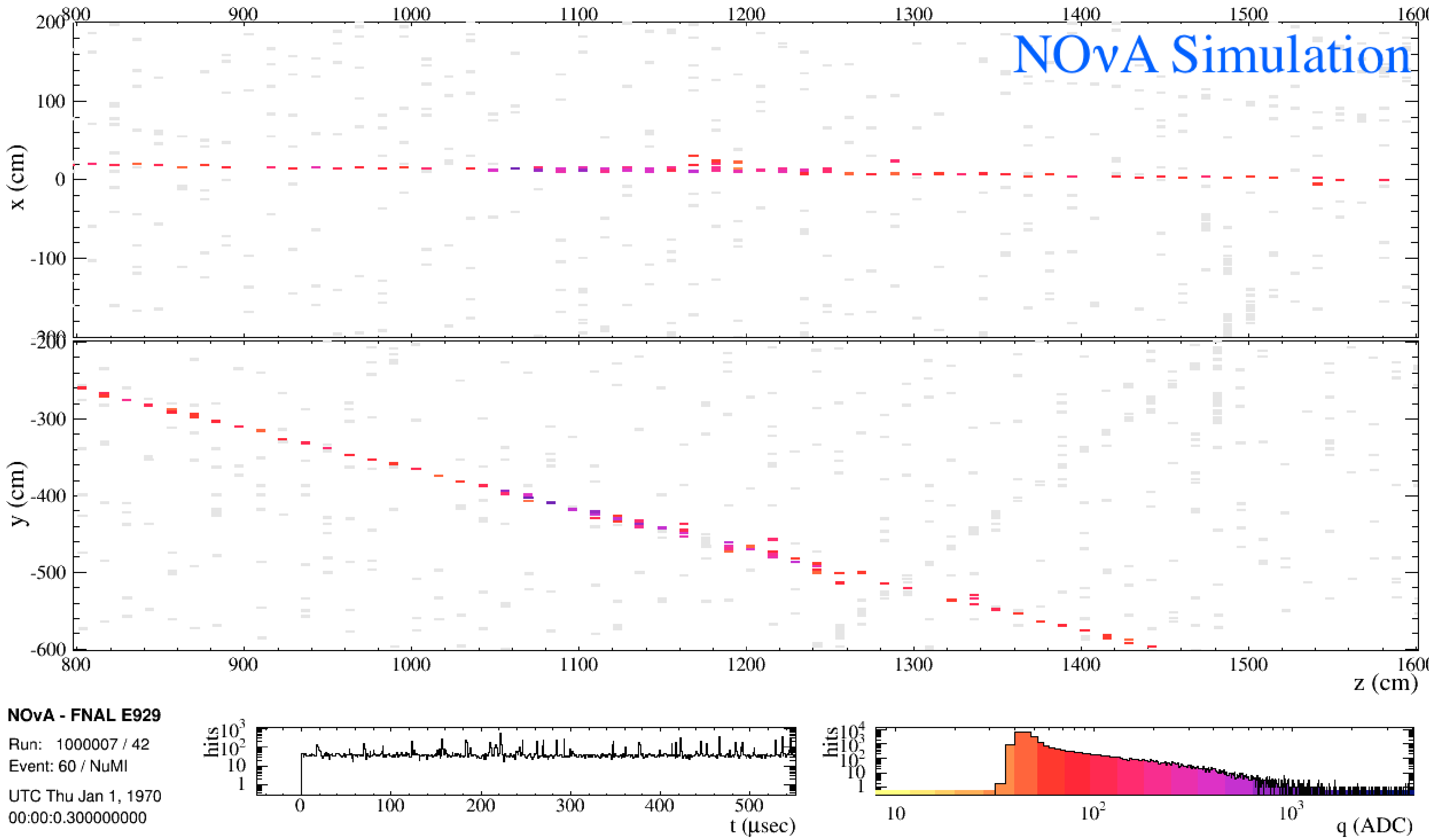
Excellent data/MC agreement in MRE sample. Efficiency difference <1%. Smaller than for traditional PIDs:

PID	Sample	Preselection	PID	Efficiency	Efficiency diff %
CVN	Data	262884	188809	0.718222	-0.36%
	MC	277320	199895	0.720809	
LEM	Data	262884	153599	0.584284	-0.73%
	MC	277320	163218	0.588555	
LID	Data	262884	175492	0.667564	2.09%
	MC	277320	181267	0.653638	

NOvA Preliminary

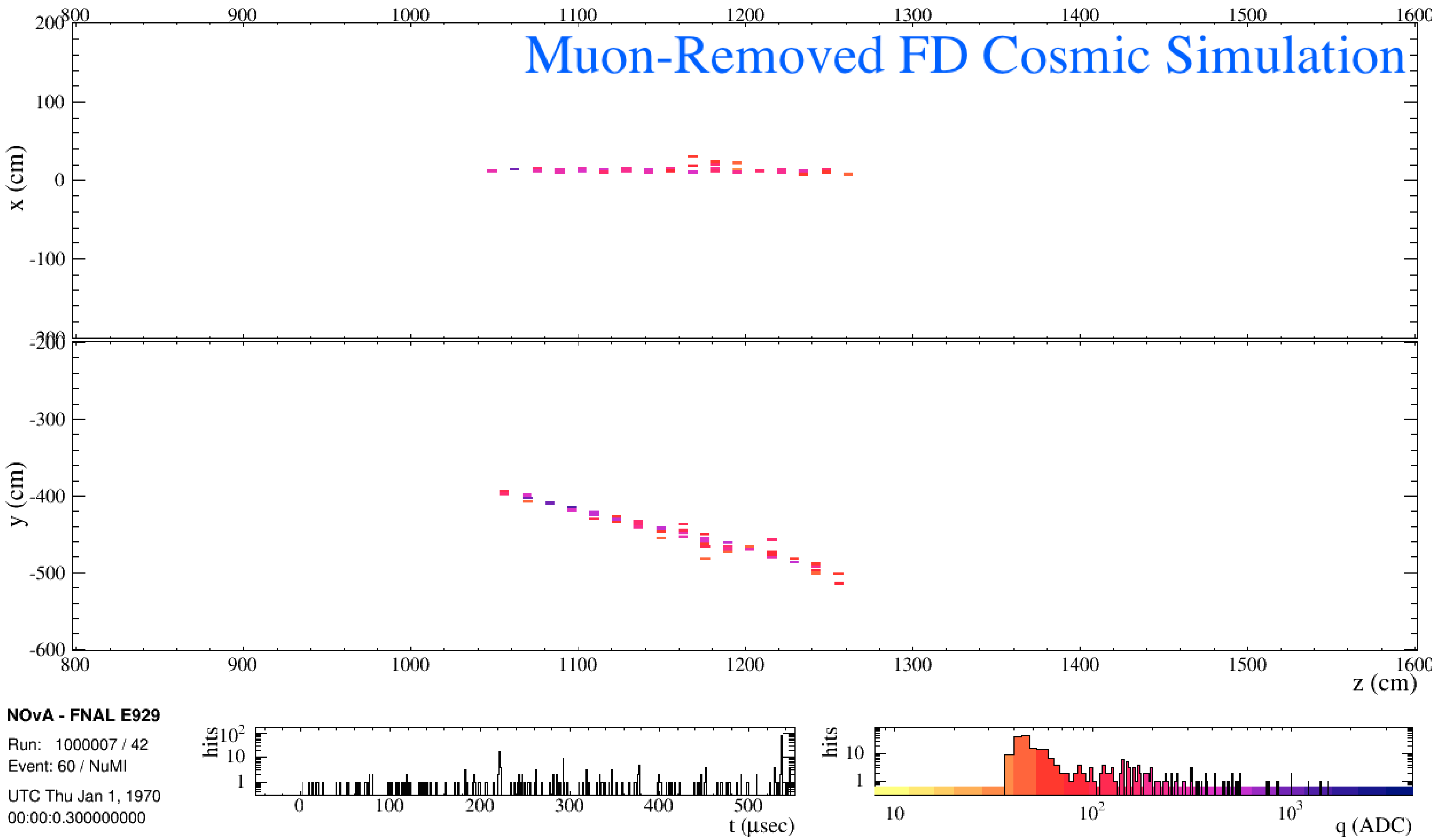


Data Driven Cross Checks: Muon Removed Bremsstrahlung



But what about the Far Detector? Try using cosmogenic activity. We find Bremsstrahlung, remove the associated muon, and see what CVN does in data vs. simulation.

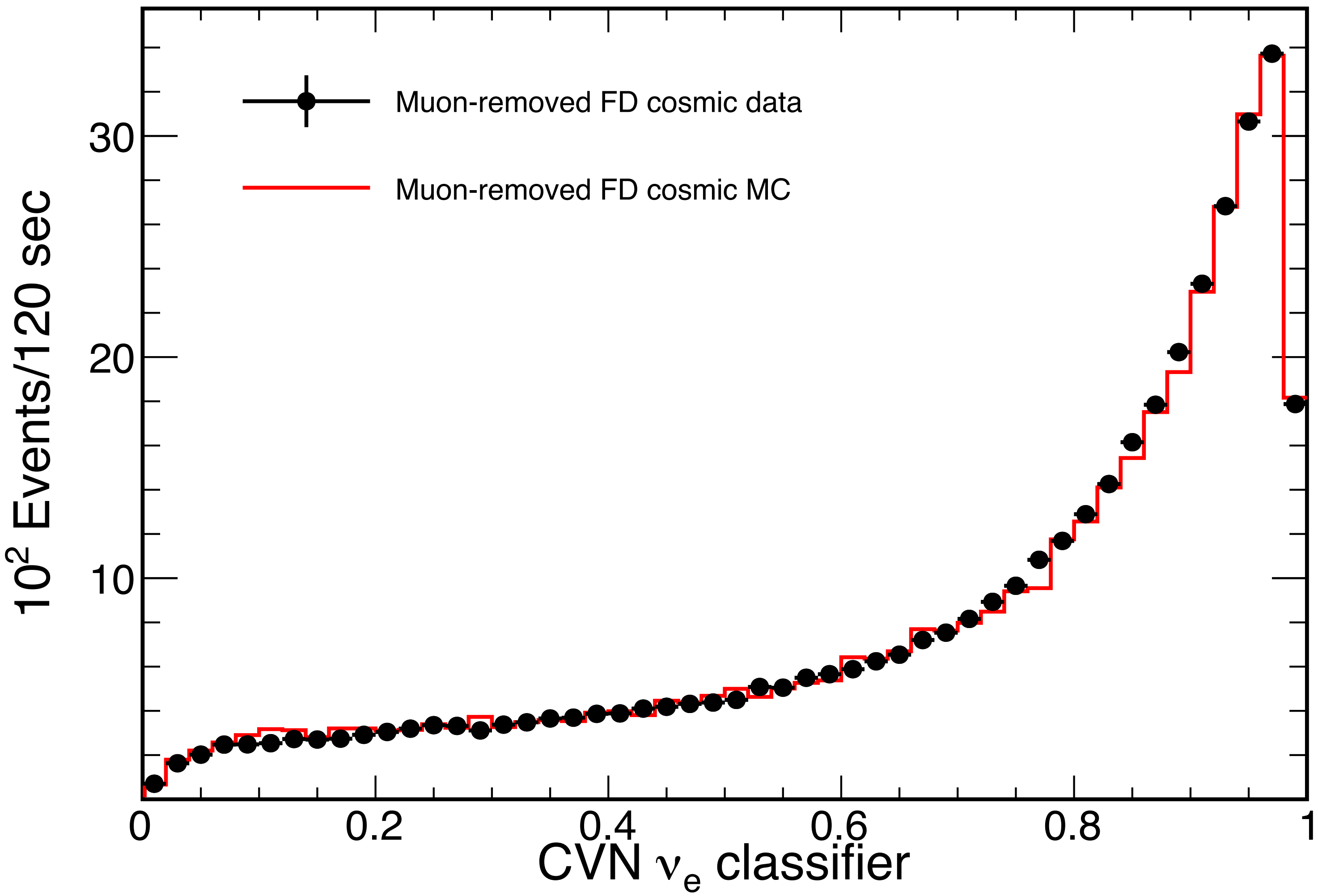
Data Driven Cross Checks: Muon Removed Bremsstrahlung



But what about the Far Detector? Try using cosmogenic activity. We find Bremsstrahlung, remove the associated muon, and see what CVN does in data vs. simulation.

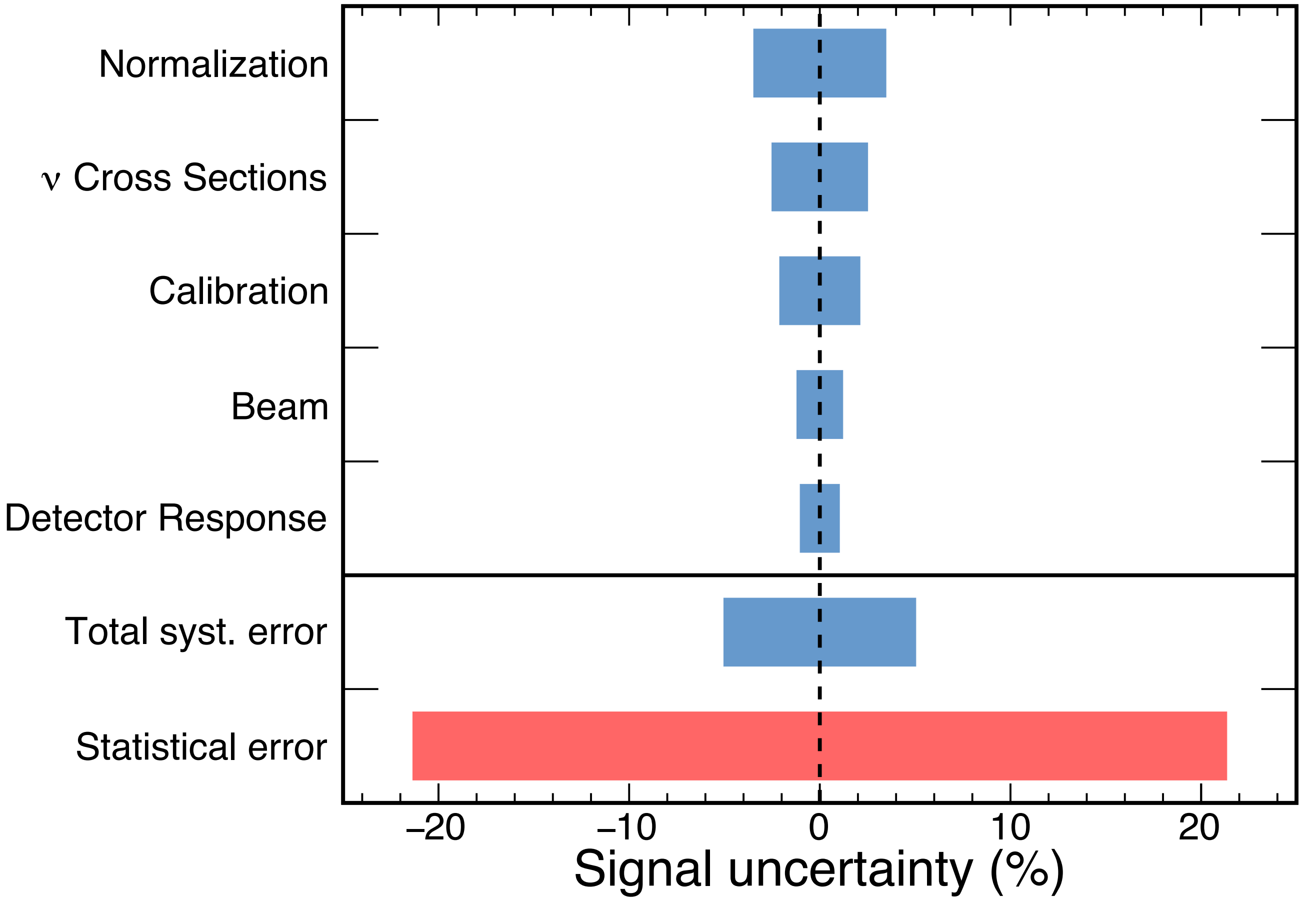
Data Driven Cross Checks: Muon Removed Bremsstrahlung

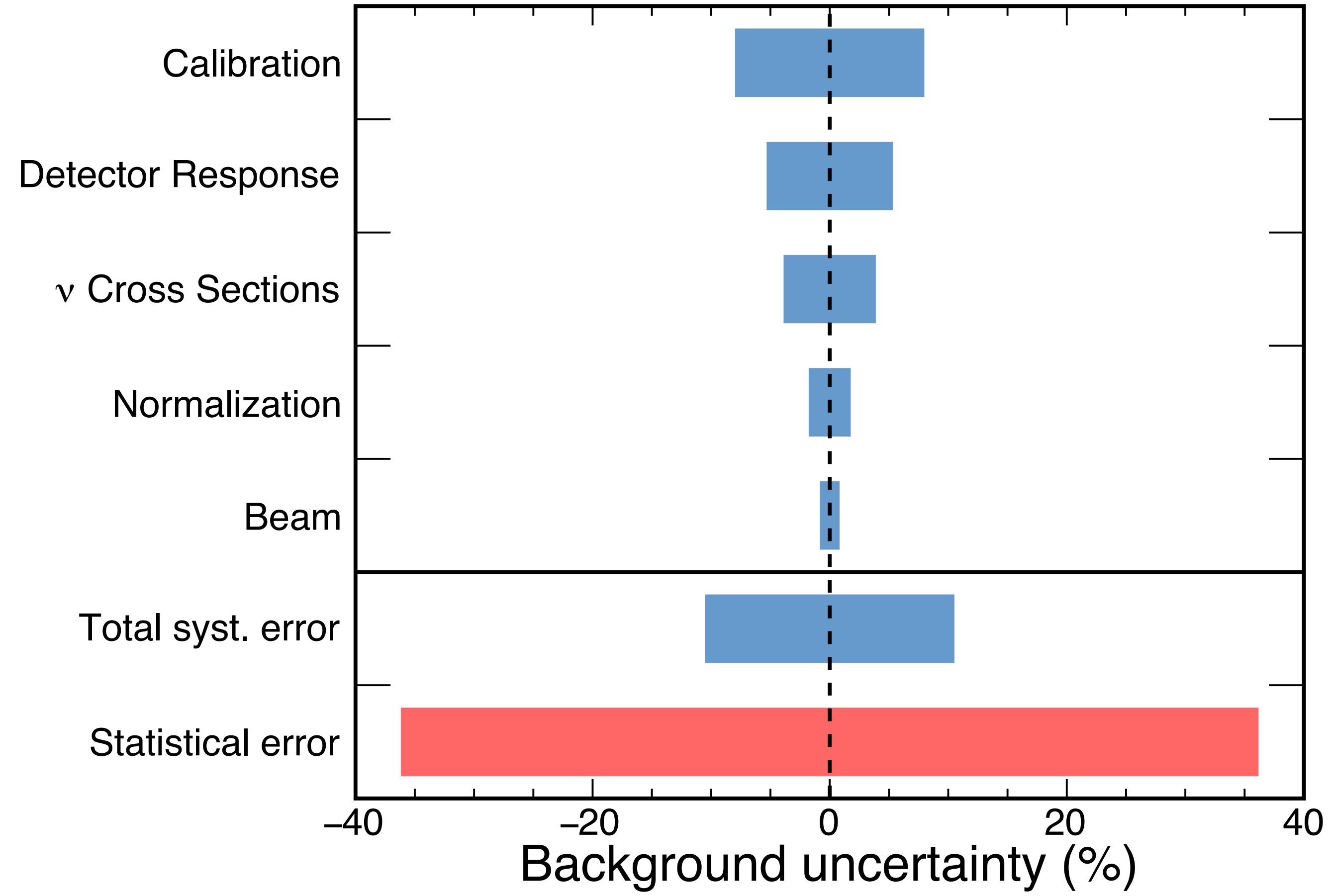
NOvA Preliminary



But what about the Far Detector? Try using cosmogenic activity. We find Bremsstrahlung, remove the associated muon, and see what CVN does in data vs. simulation.

Simulation Cross Checks: Systematic Studies

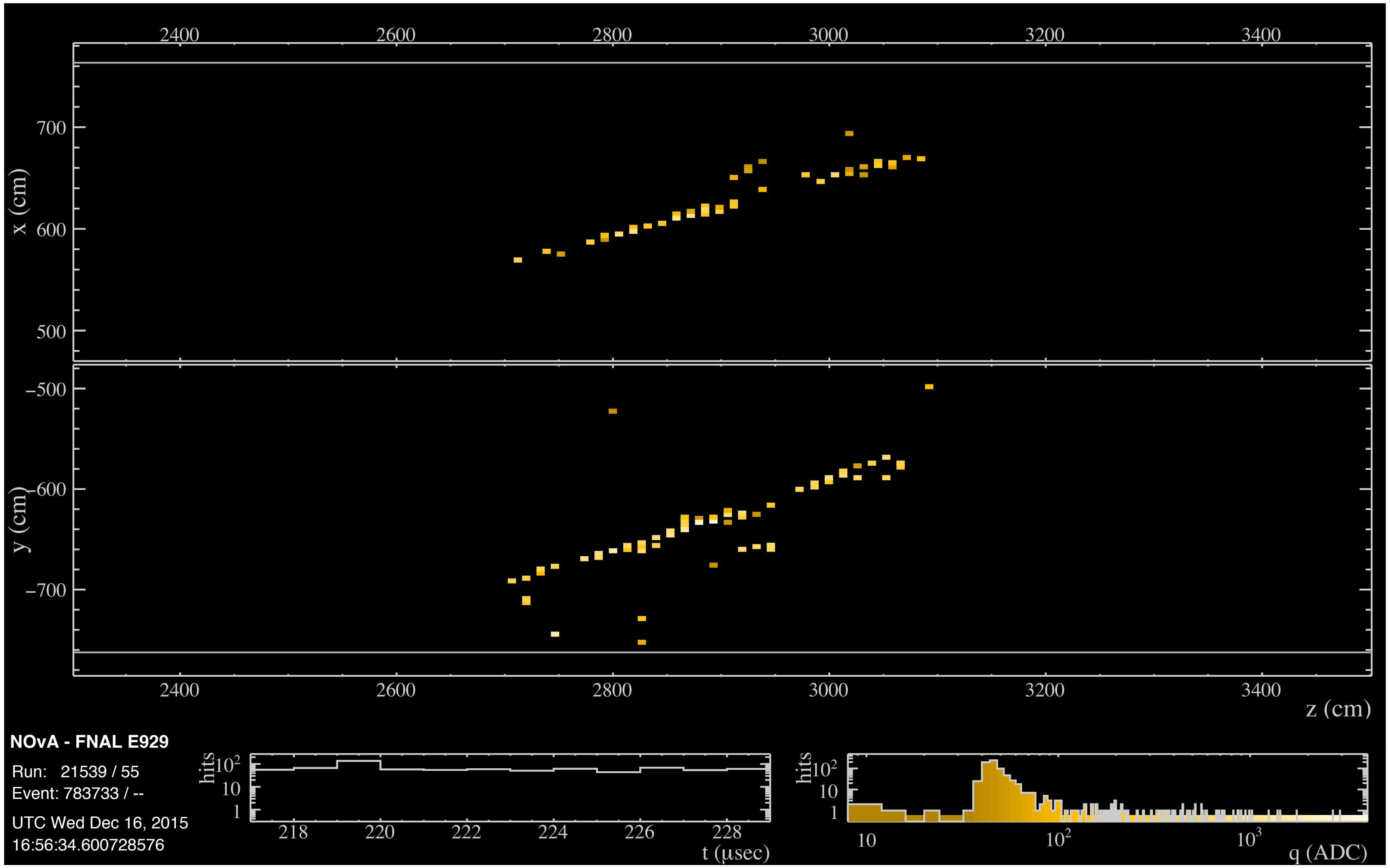




We also explore a number of simulated uncertainties, propagating changes all the way to the final predicted rate of signal and background events at the far detector.

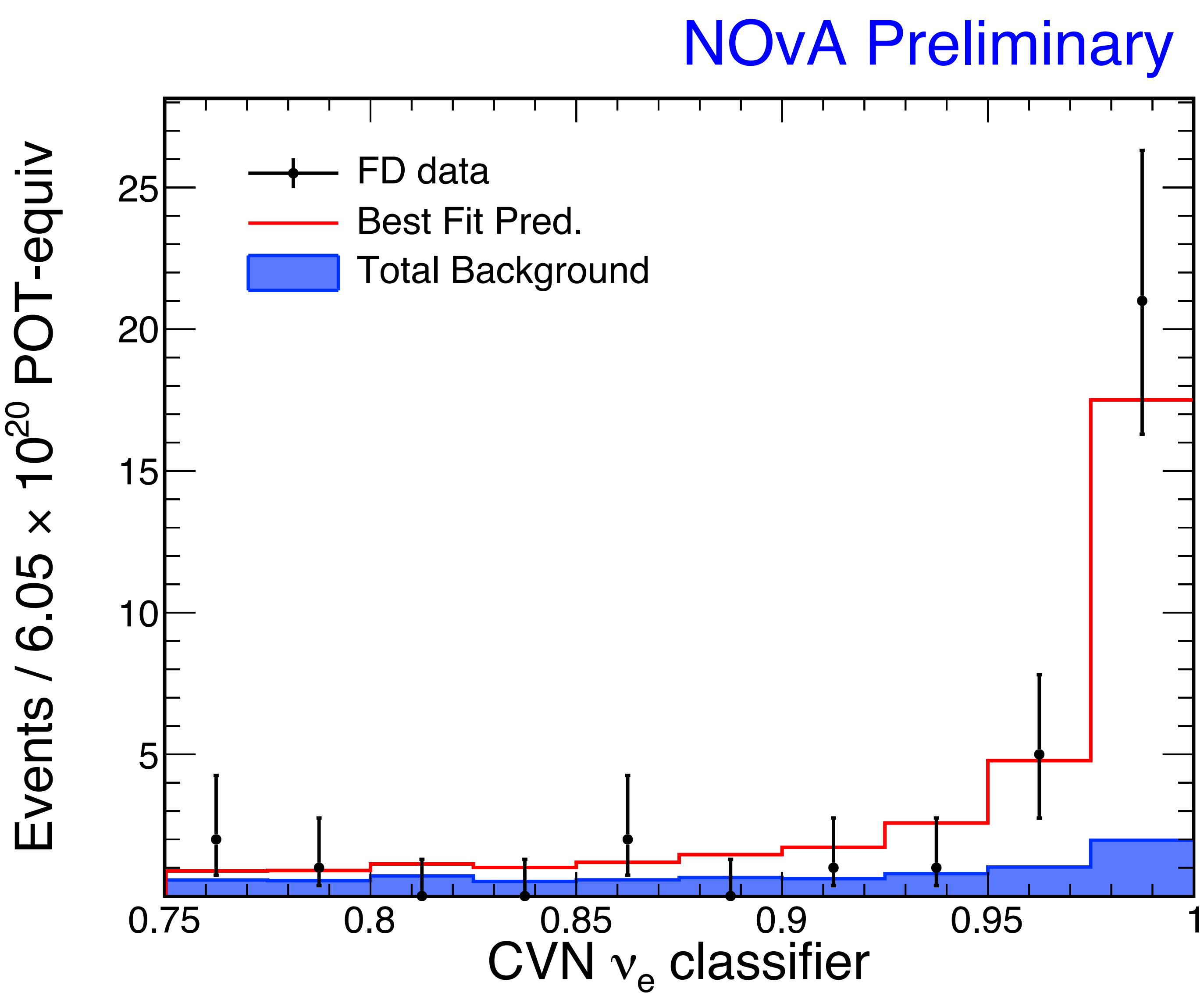
Final sys. error: ~5% on signal and ~10% on the background.

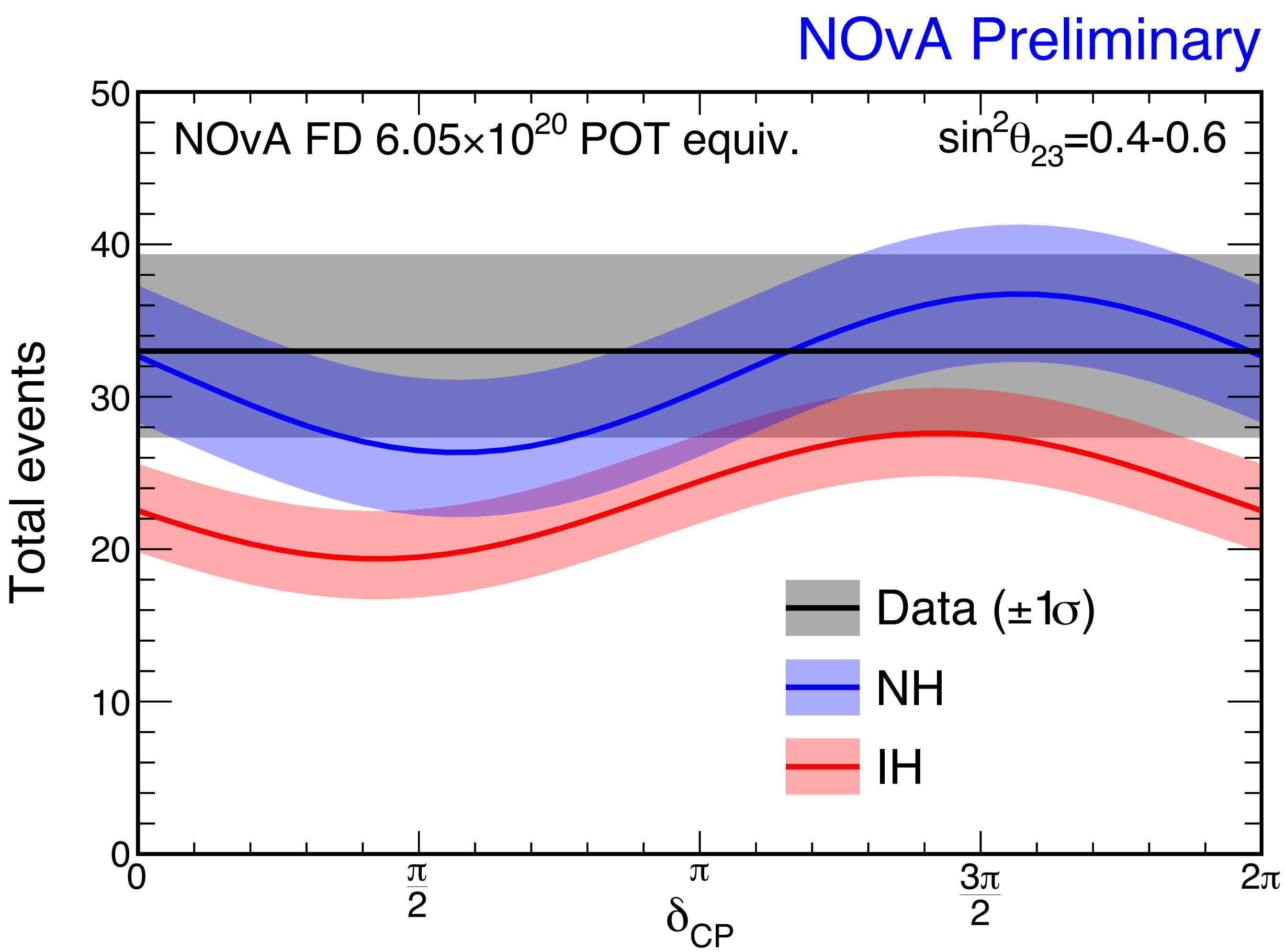
Far Detector Data



After exhaustive cross checks we opened the box.
Excellent data/MC agreement in the 33 selected events across the board!

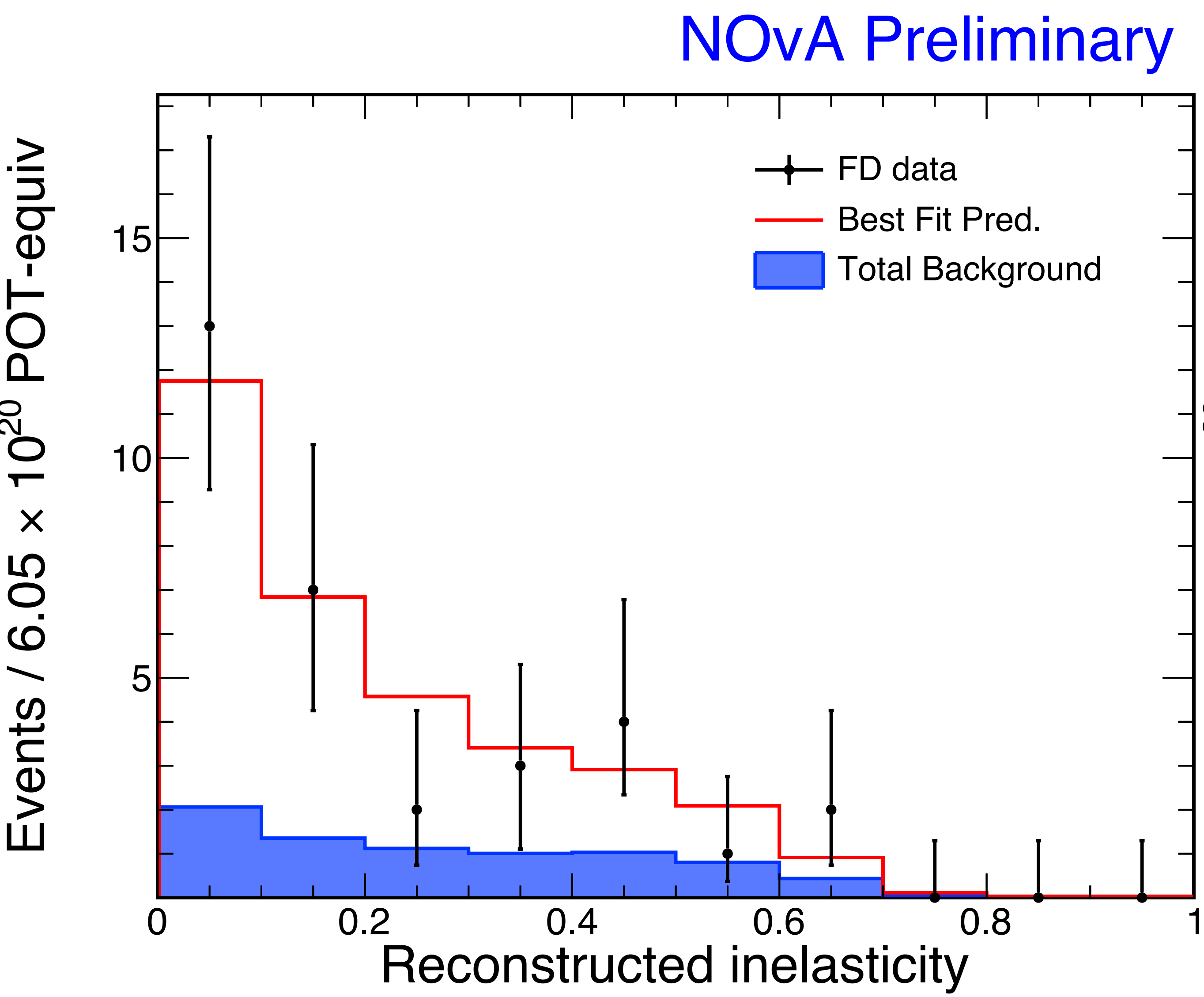
Far Detector Data

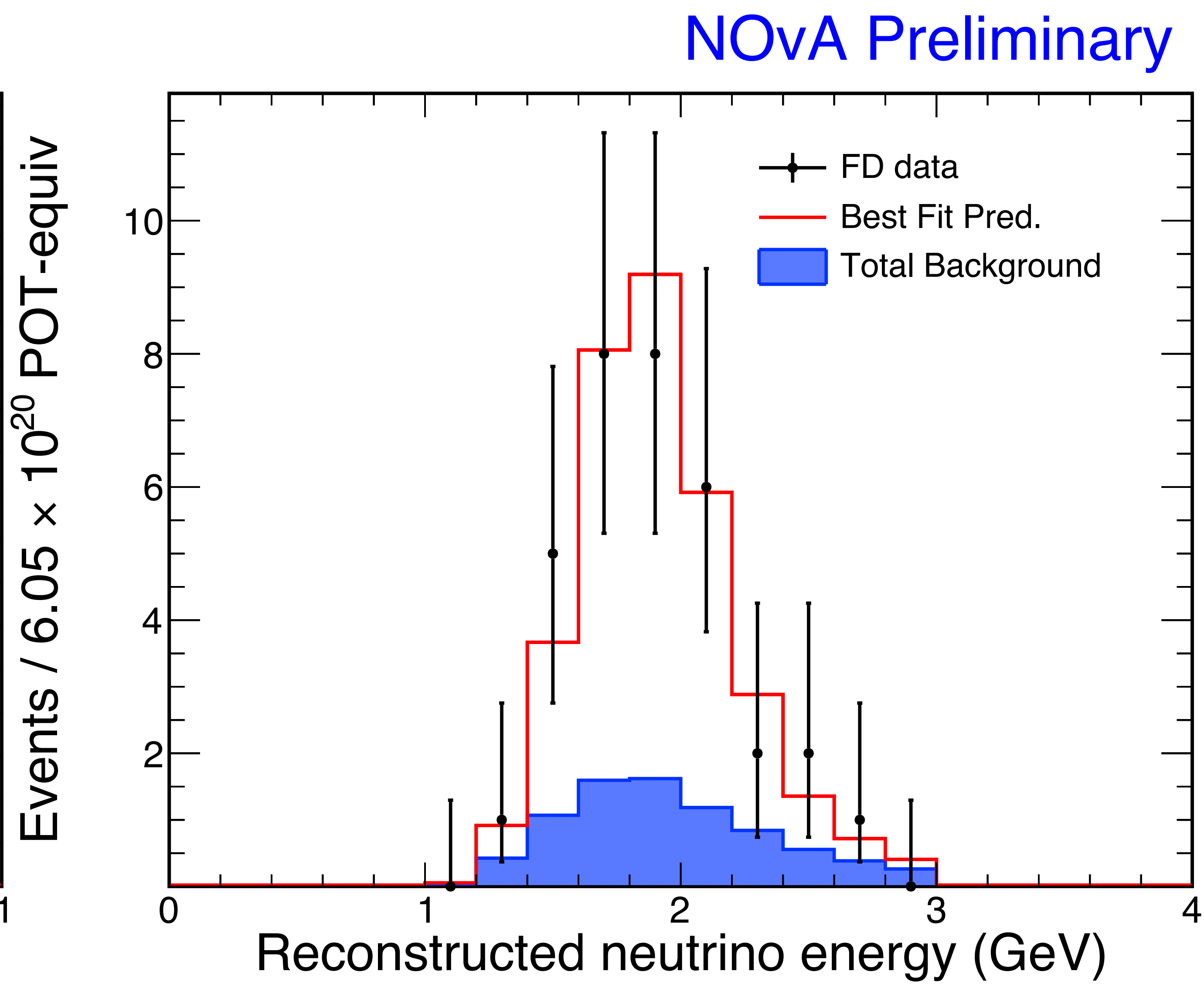




After exhaustive cross checks we opened the box.
Excellent data/MC agreement in the 33 selected events across the board!

Far Detector Data



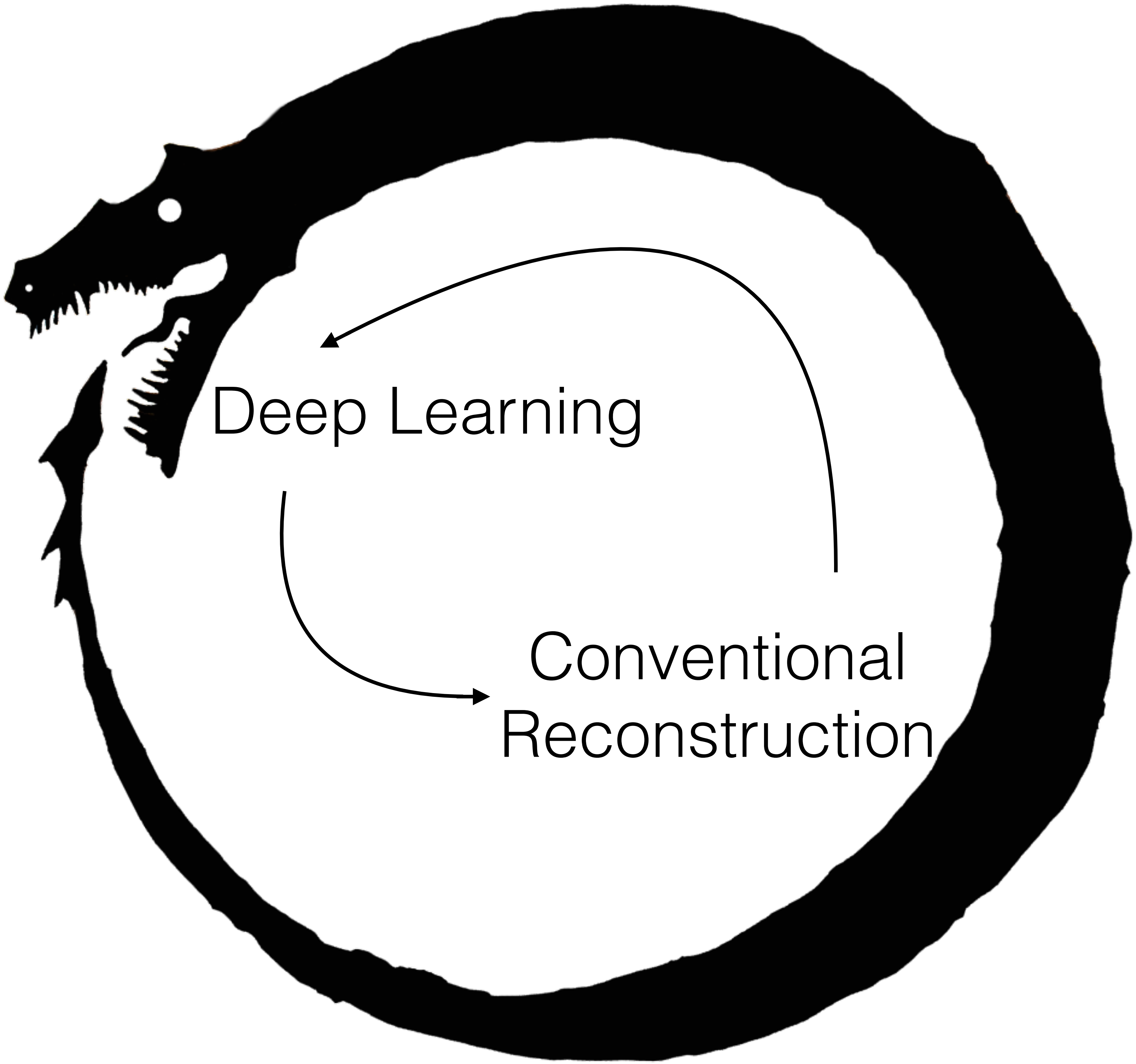


After exhaustive cross checks we opened the box. Excellent data/MC agreement in the 33 selected events across the board!

The Future

The original dream

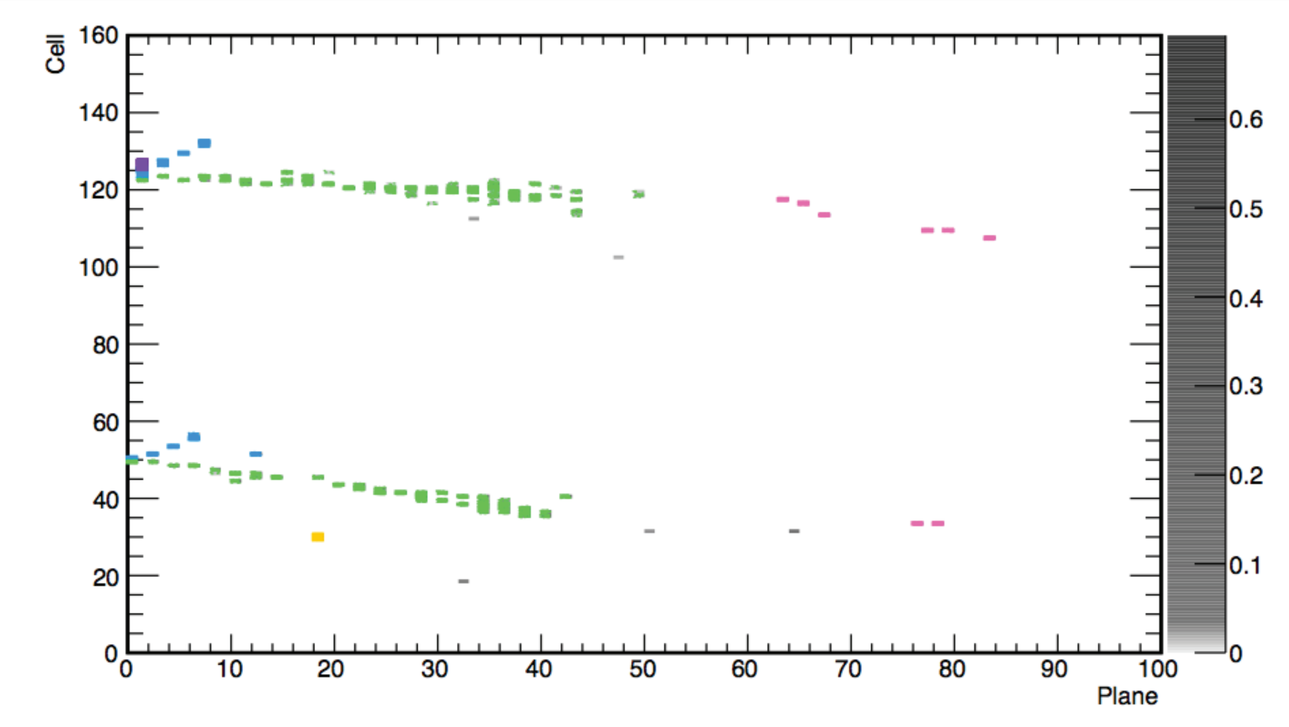
Where we're really going



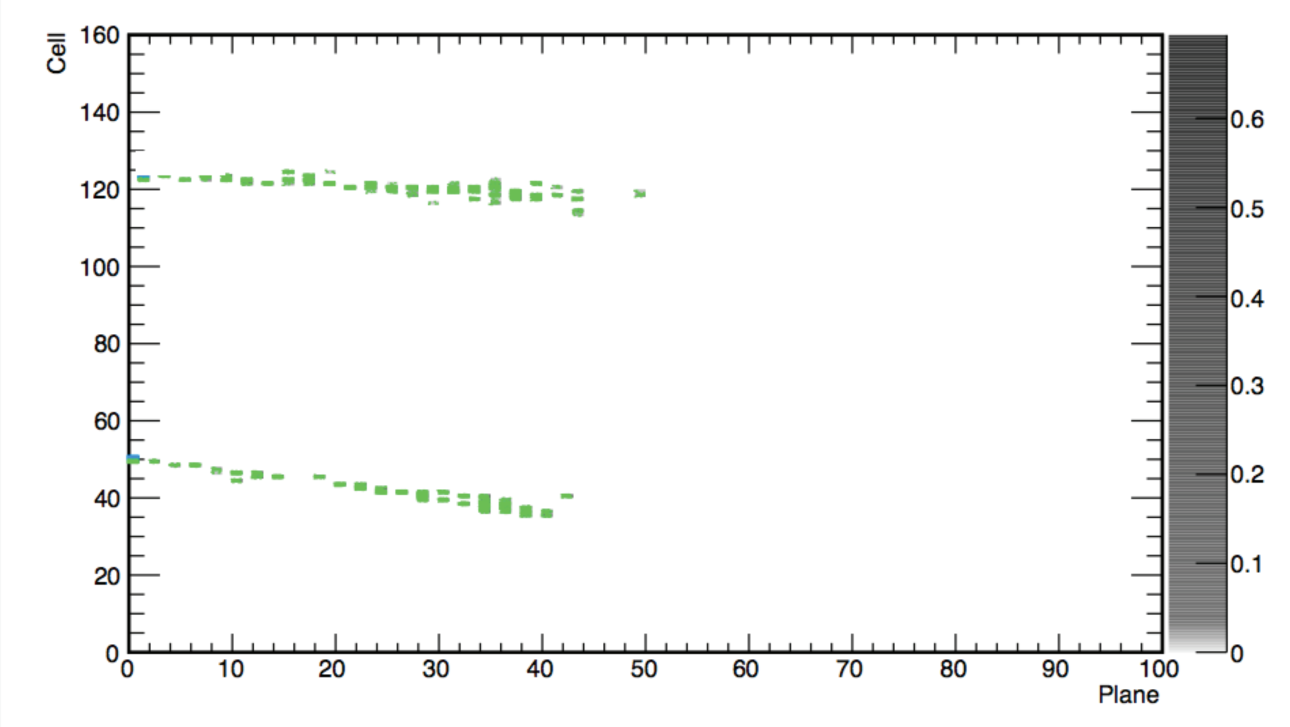
Prong ID

Can we use CVN to ID our reconstructed objects, like showers?

CVN Input: 4 views



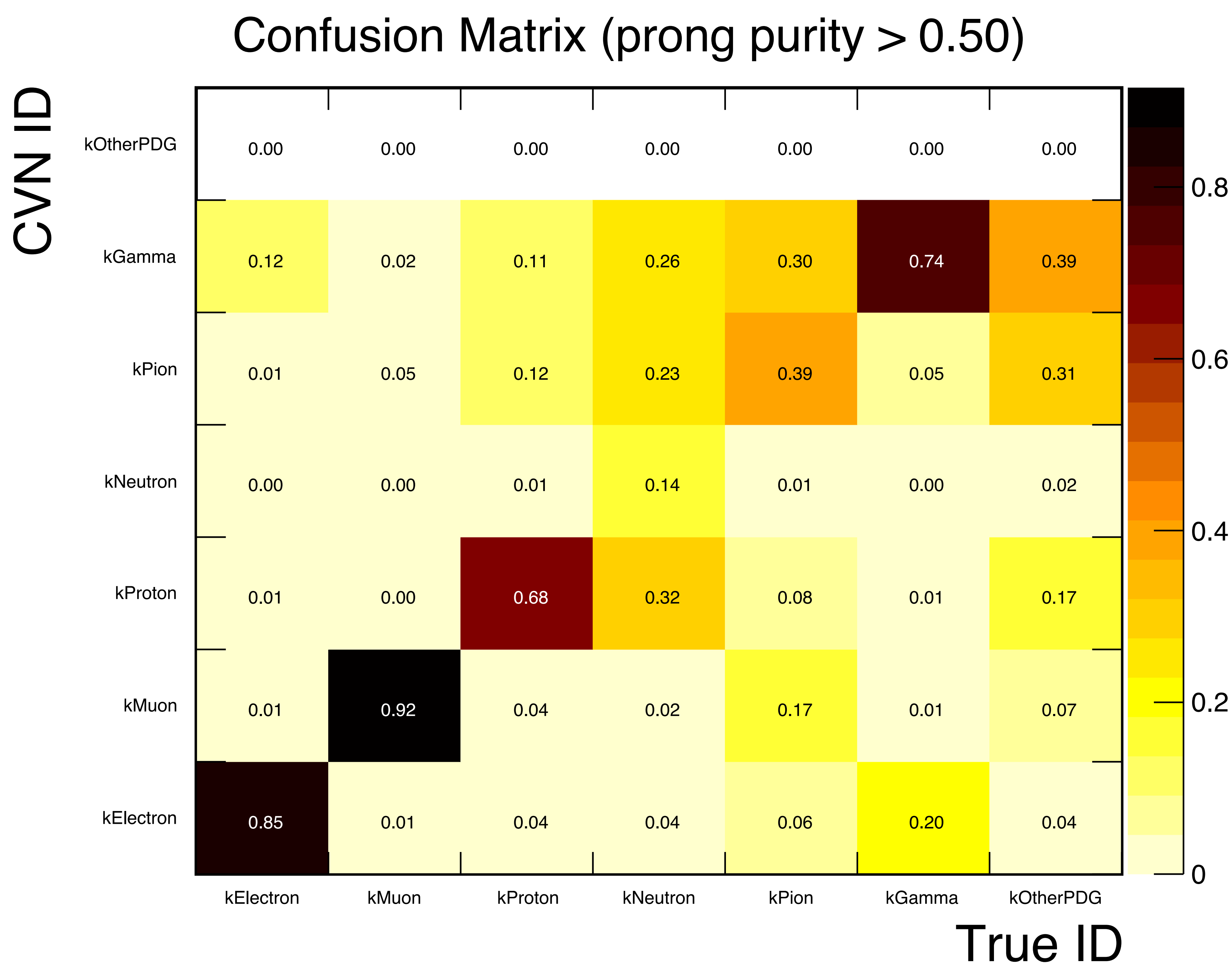
2 complete views



2 prong views

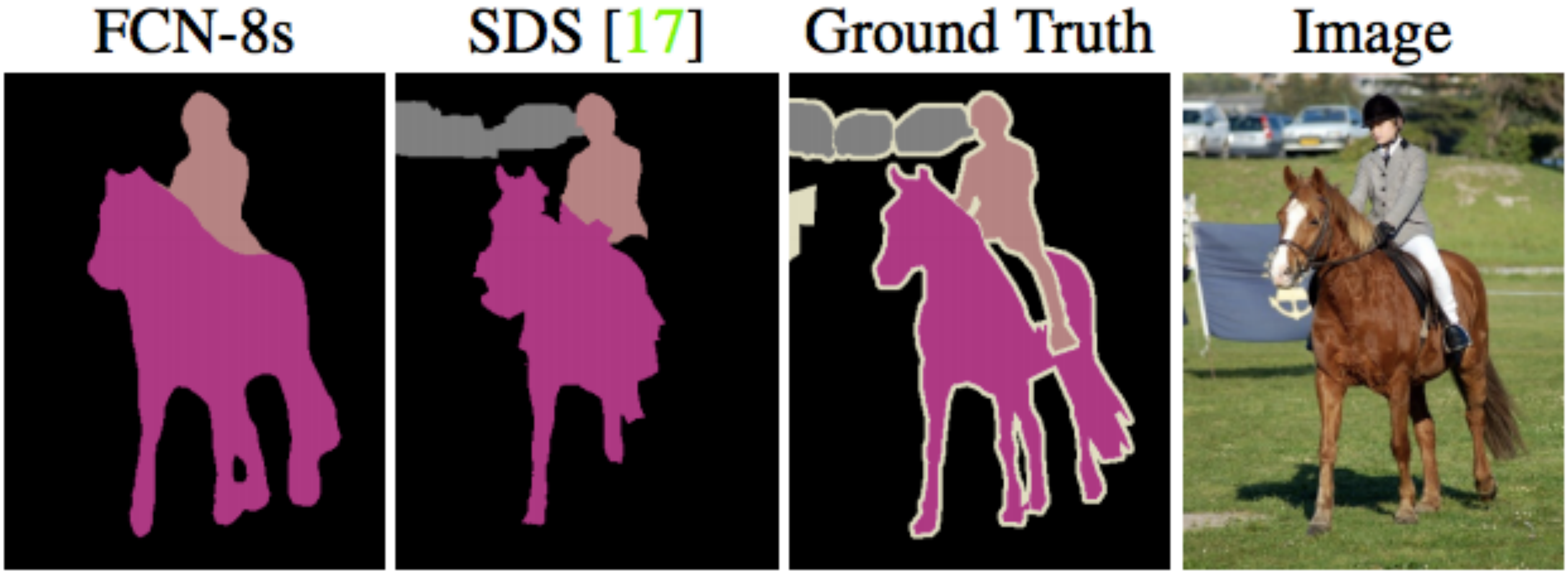
Prong ID

Very promising! Why stop at reconstructed showers though?



Semantic Segmentation

Semantic segmentation takes advantage of information at every lay of a CNN to perform a identification at the pixel level.

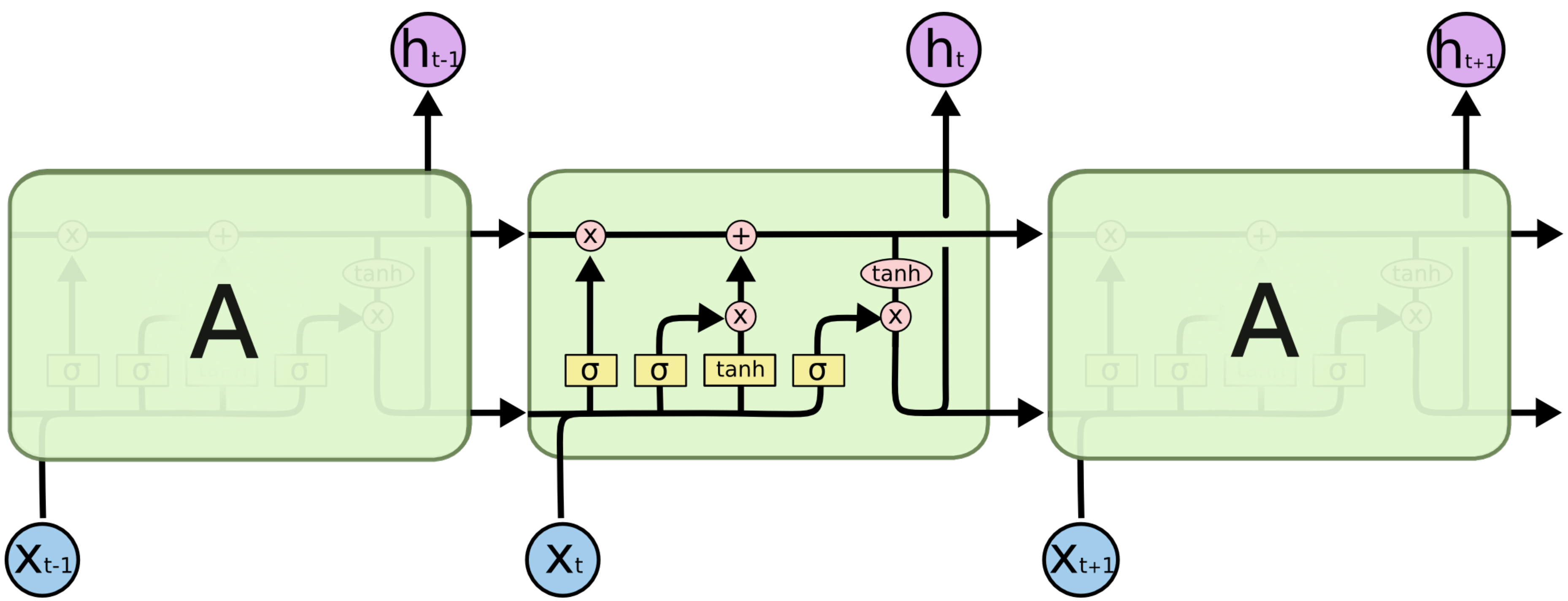


http://www.cs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

Recurrent Neural Networks

"Recurrent" Neural Networks are a structure for parsing sequential information through a recurrent network structure.

Great at learning patterns, very successful in translation tasks. Could be powerful for signal processing, more efficient "image" parsing, or online triggering?



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

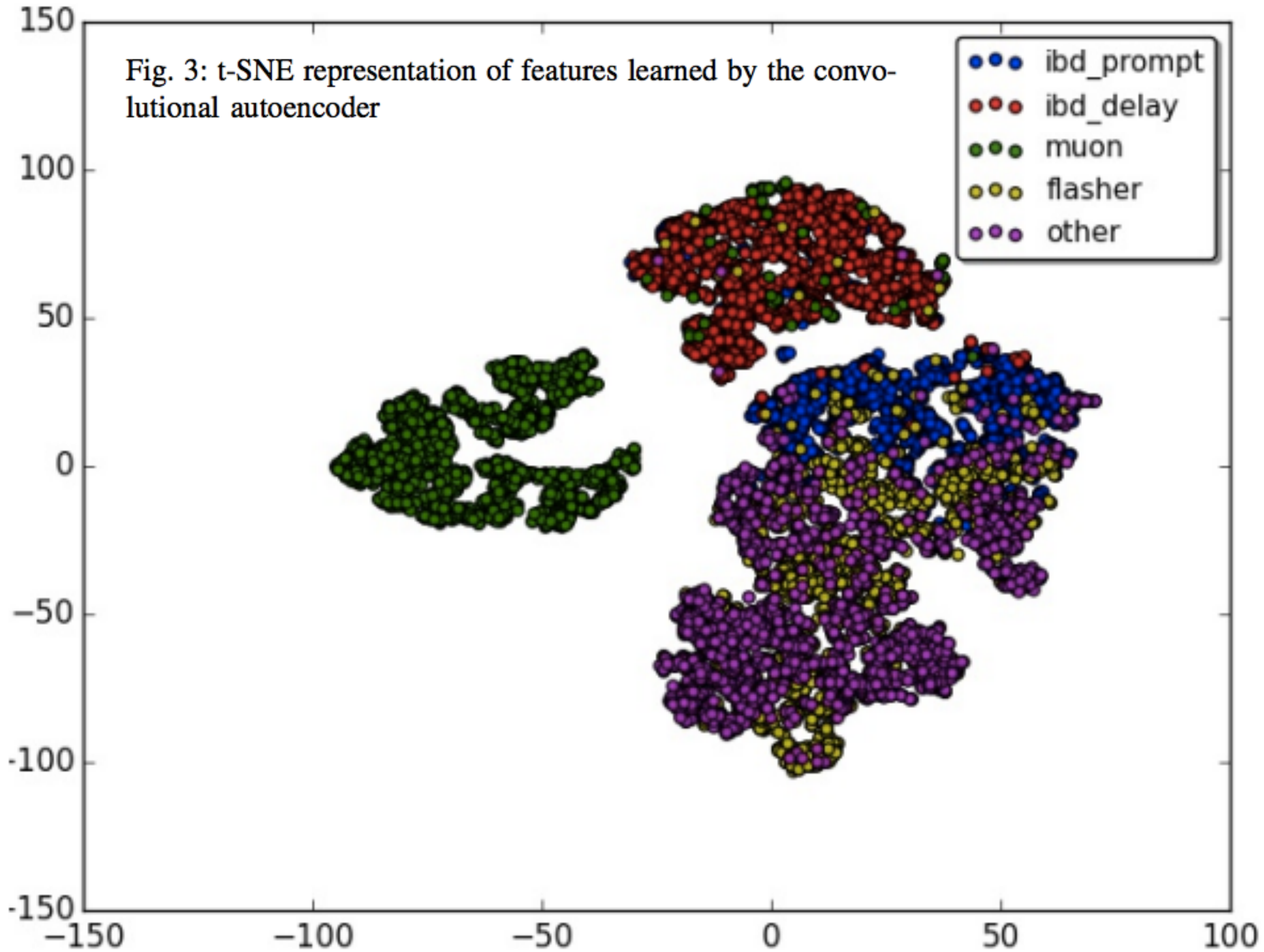
Adversarial Networks

Add nuance to your training by pitting it against an adversarial network which pushes optimization in another direction.

Examples of exciting Deep Learning + science research here, including recent attempts to train out bias to nuisance parameters:
"Learning to Pivot with Adversarial Networks"
arXiv:1611.01046

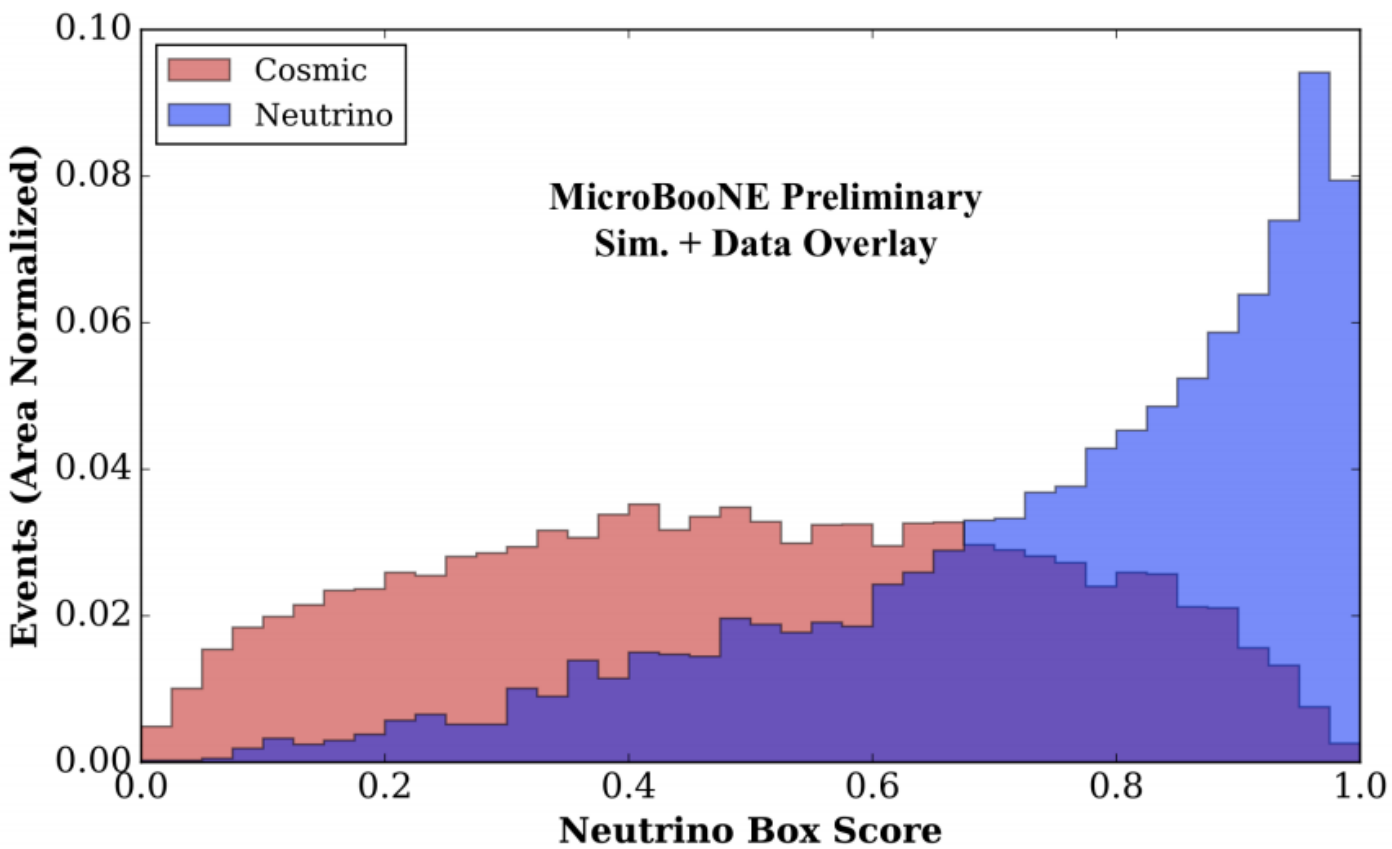
An Active Field!

Daya Bay



arXiv:1601.07621

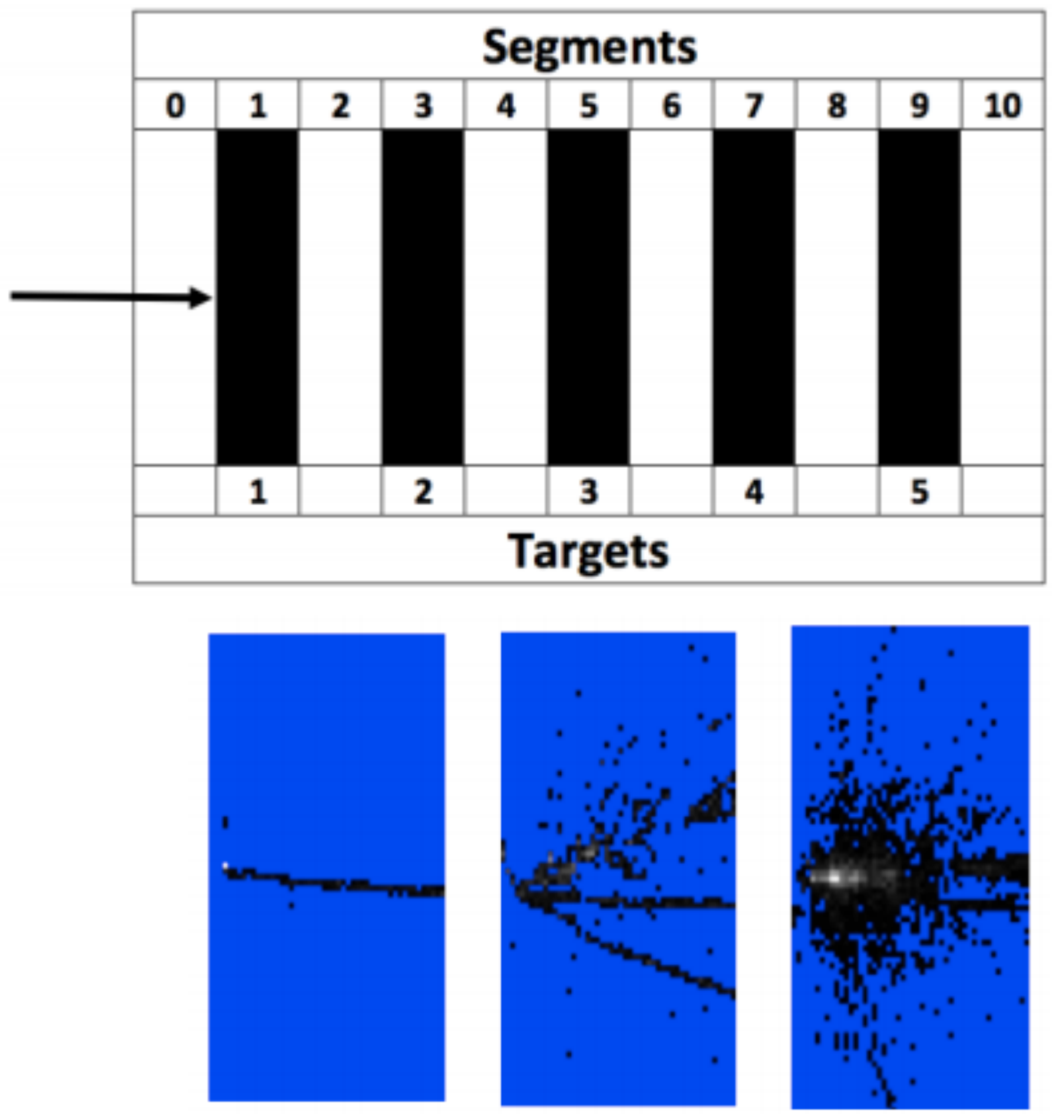
MicroBooNE



MicroBooNE-NOTE-1019-PUB

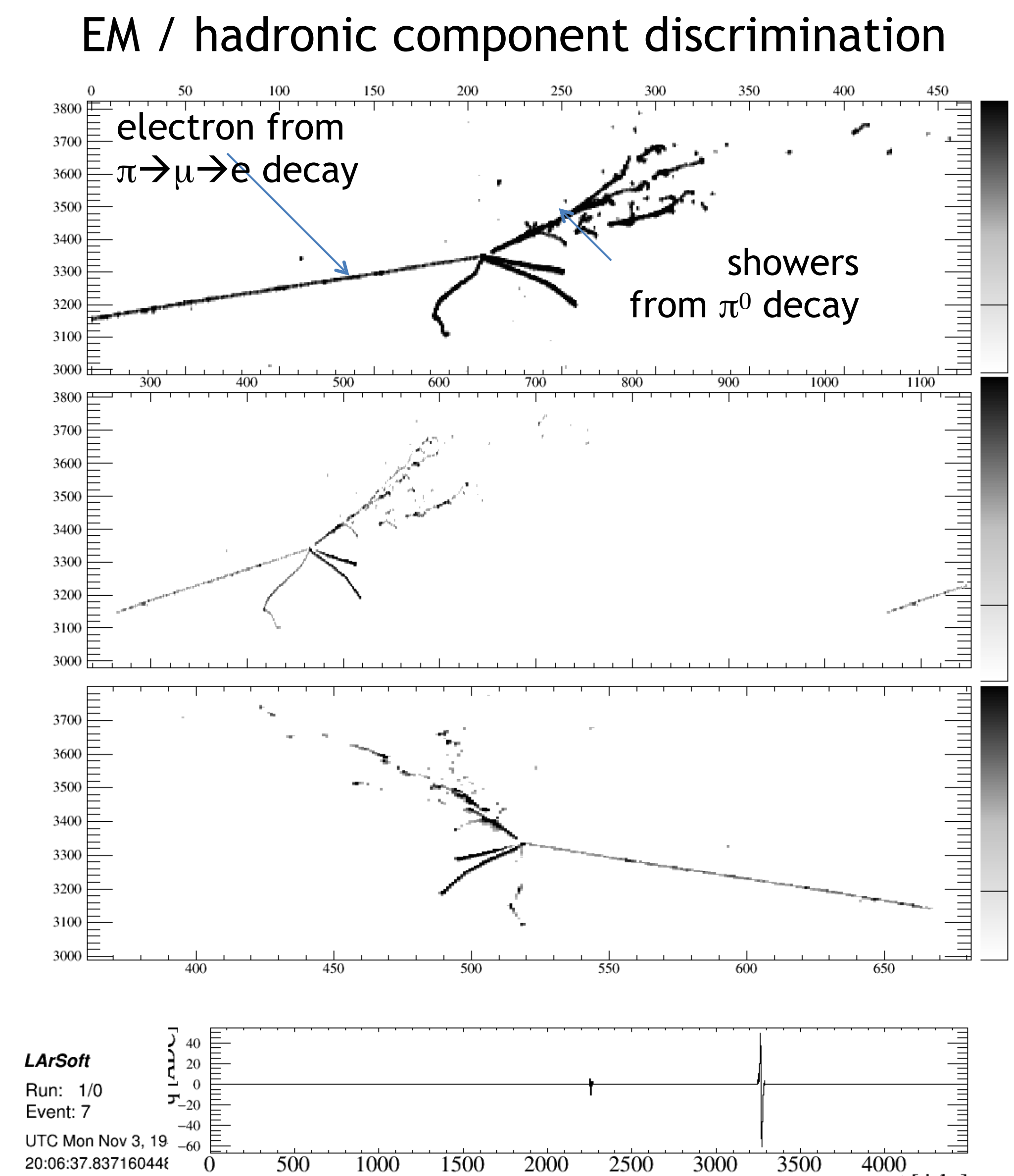
An Active Field!

MINERvA and MENNDL



PhyStat-nu Fermilab 2016 (19-September 21, 2016)

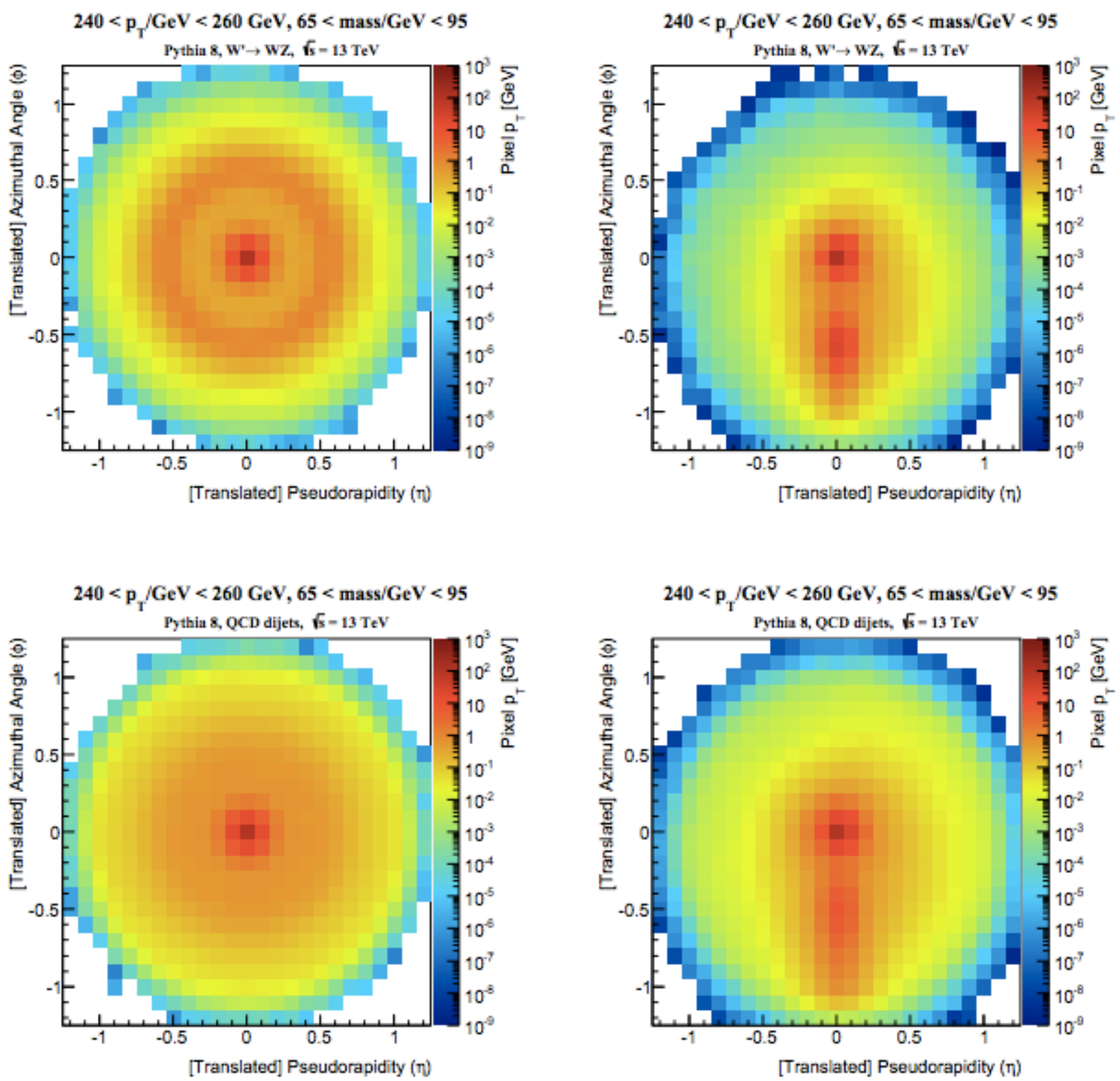
Lariat and ProtoDUNE



Private communication, Robert Sutlej

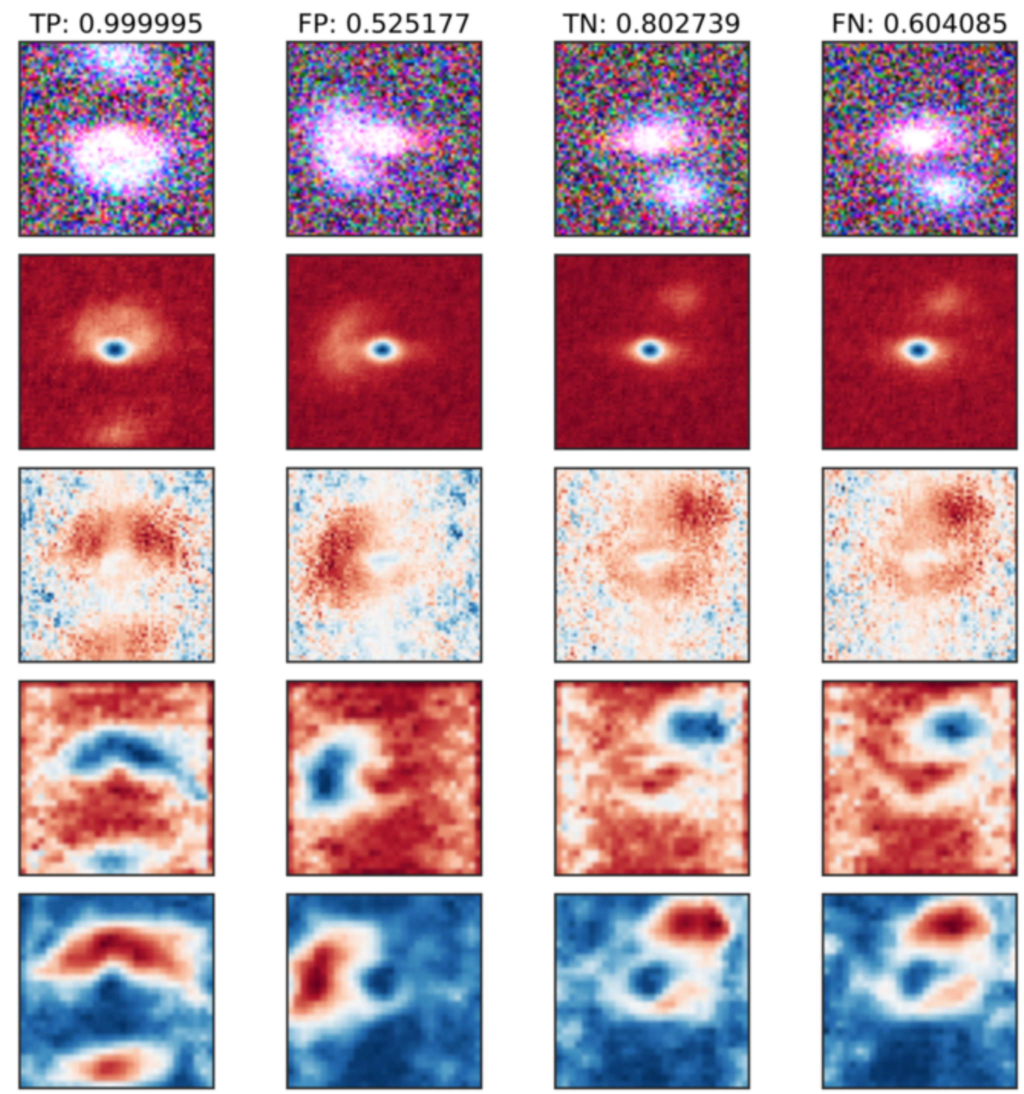
An Active Field!

ATLAS:



arXiv:1511.05190

DES:



Private communication, Brian Nord in the Dark Energy Survey team

Conclusions

The first high energy particle physics measurement to use deep learning! Already also used in the first NOvA sterile neutrino analysis and being explored as an option for a number of other searches.

Just the tip of the iceberg! Huge amounts of room to optimize our classification network, and to explore other applications of deep learning tools.

While you wait you should check out our recent paper:
"A Convolutional Neural Network Neutrino Event Classifier"
A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, P. Vahle
https://arxiv.org/abs/1604.01444
Journal of Instrumentation, Volume 11, September 2016

Join the conversation at https://hepmachinelearning.slack.com/

Q&A

Many thanks to the NOvA collaboration, Fermilab National Accelerator laboratory, and to the National Science Foundation.

HEP Deep Learning Papers

“A Convolutional Neural Network Neutrino Event Classifier” **JINST 11 (2016) no. 09, P09001**

“Revealing Fundamental Physics from the Daya Bay Neutrino Experiment using Deep Neural Networks” **arXiv:1601.07621**

“Background rejection in NEXT using deep neural networks” **arXiv:1609.06202**

“Searching for exotic particles in high-energy physics with deep learning” **Nature Communications 5, Article number: 4308 (2014)**

“Jet-Images -- Deep Learning Edition” **arXiv:1511.05190**

“Jet Substructure Classification in High-Energy Physics with Deep Neural Networks” **Phys.Rev. D93 (2016) no.9, 094034**

“Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks” **arXiv:1609.00607**

“Machine learning techniques in searches for $t\bar{t}h$ in the $h \rightarrow b\bar{b}$ decay channel” **arXiv:1610.03088**

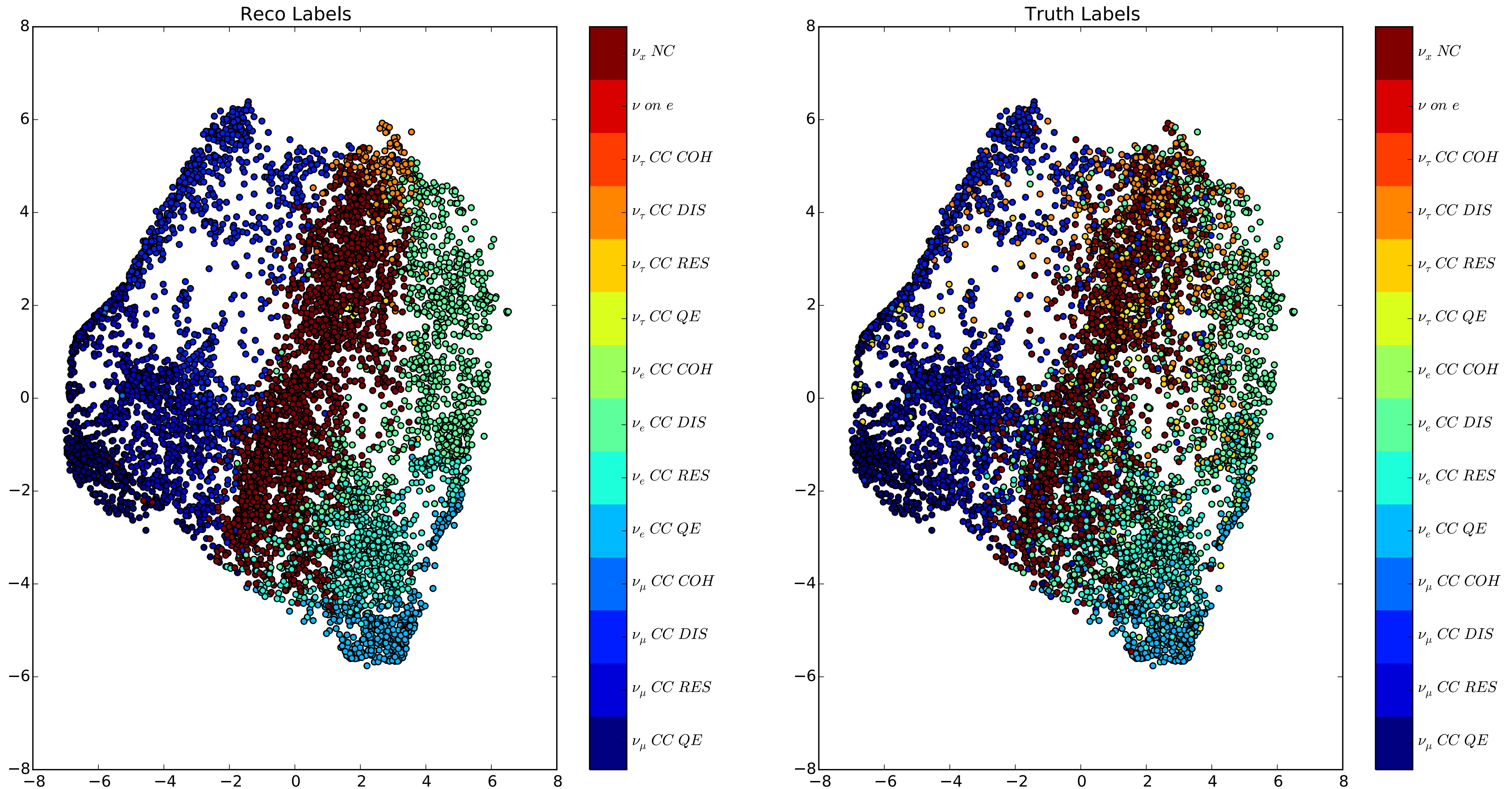
“Learning to Pivot with Adversarial Networks” **arXiv:1611.01046**

“Convolutional Neural Networks Applied to Neutrino Events in a Liquid Argon Time Projection Chamber” **MicroBooNE-NOTE-1019-PUB**

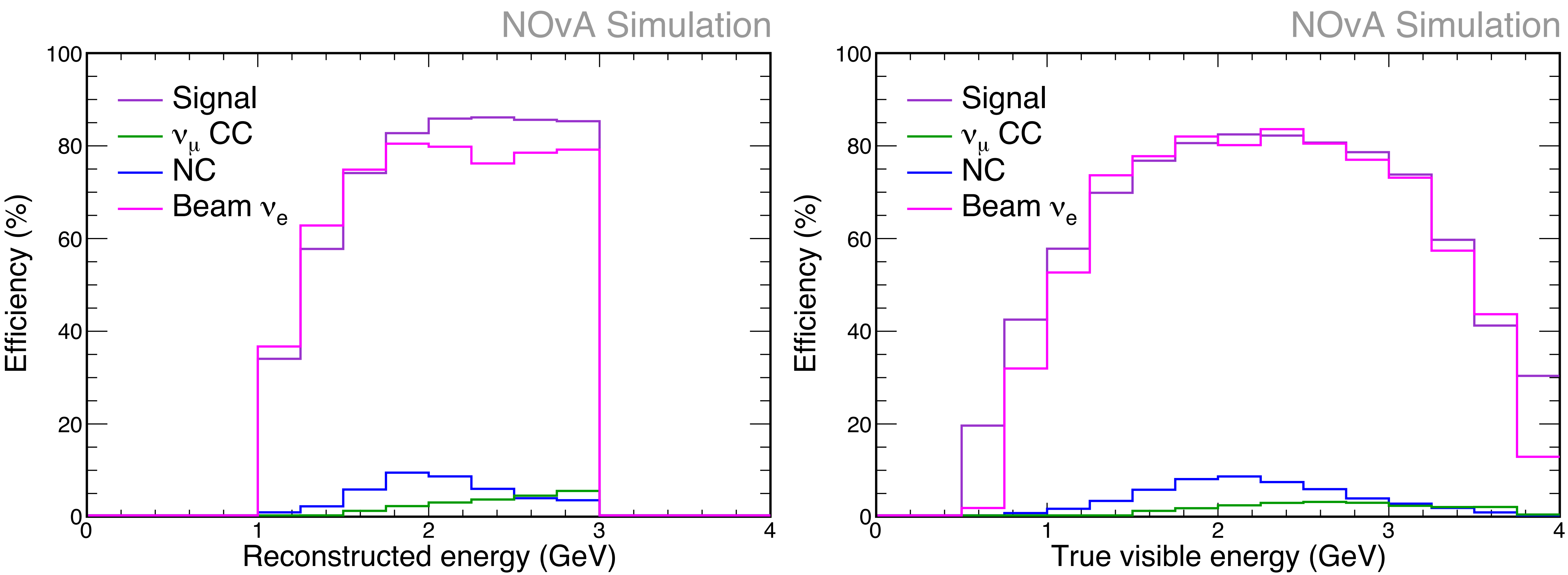
Interesting Reading

Some high level discussion of the use of these tools in science:
http://www.nature.com/news/can-we-open-the-black-box-of-ai-1.20731
I would recommend starting by reading this:
http://deeplearning.net/tutorial/lenet.html
then I'd read through the various links on the caffe homepage
Our own paper which might be of interest:
https://arxiv.org/abs/1604.01444
Some other useful papers:
http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf <-- introduces the idea of networks in networks
http://arxiv.org/pdf/1409.4842v1.pdf <-- introduces a specific google network in network called an inception module which we've found to be very powerful
http://www.cs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf <-- semantic segmentation
Developments in the googlenet, which is great way to track general trends in the field
http://arxiv.org/abs/1502.03167 <-- introduces batch normalization
http://arxiv.org/pdf/1512.00567.pdf <-- smarter kernel sizes
http://arxiv.org/abs/1602.07261 <-- introducing residual layers
Also fun though I've yet to think of a good hep application:
http://colah.github.io/posts/2015-08-Understanding-LSTMs/
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

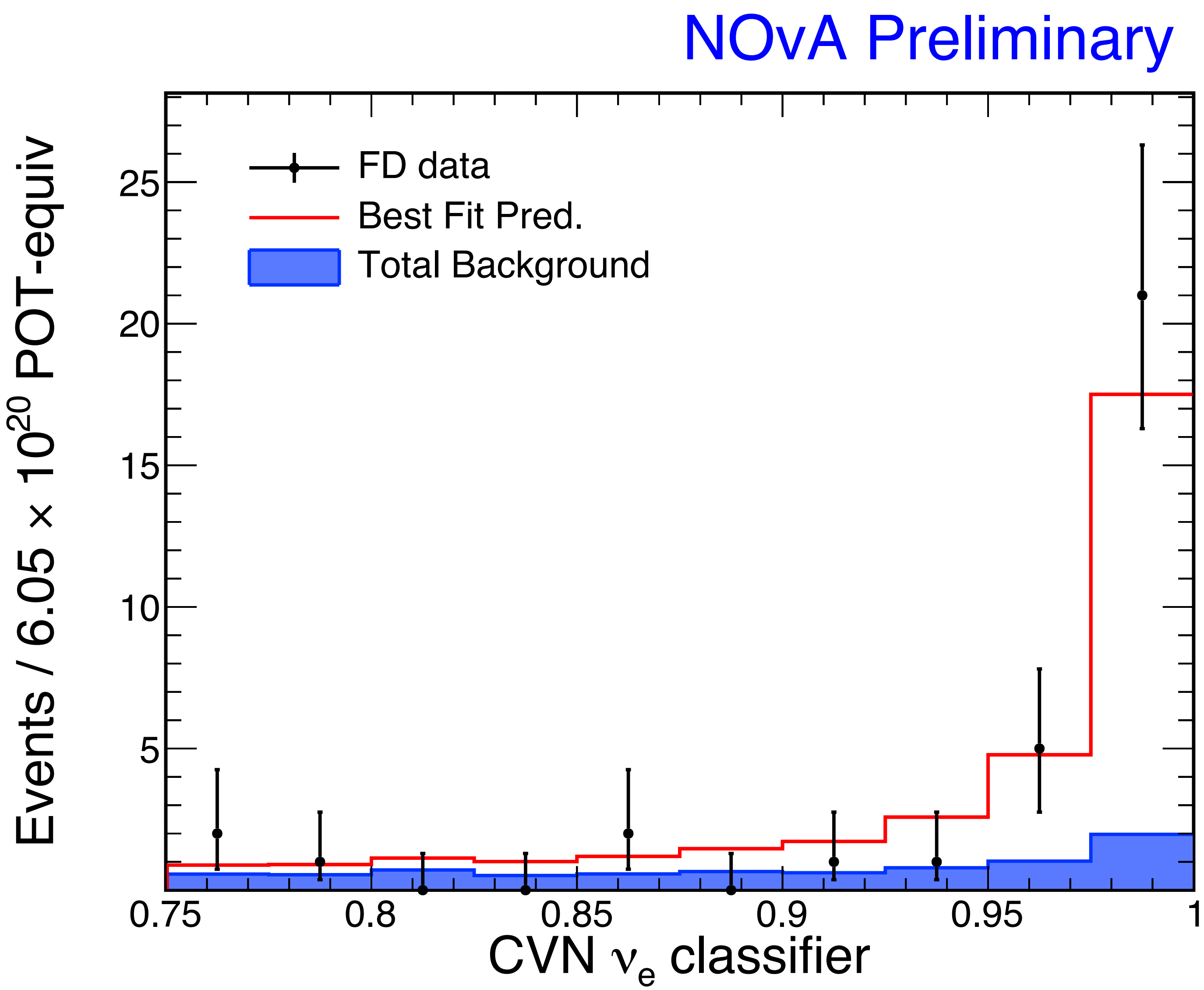
t-SNE Representation of Test Sample

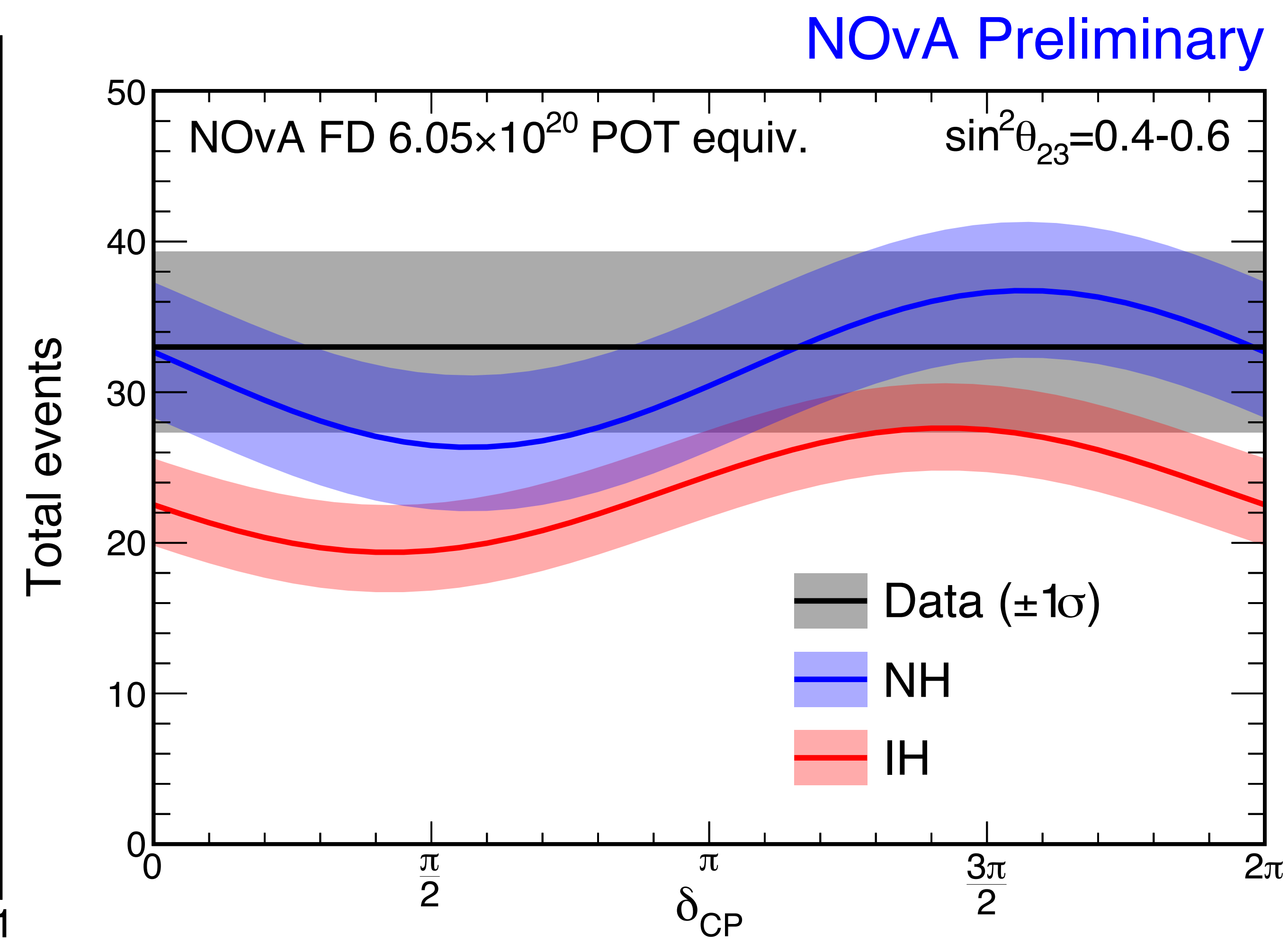


ν_e Efficiency



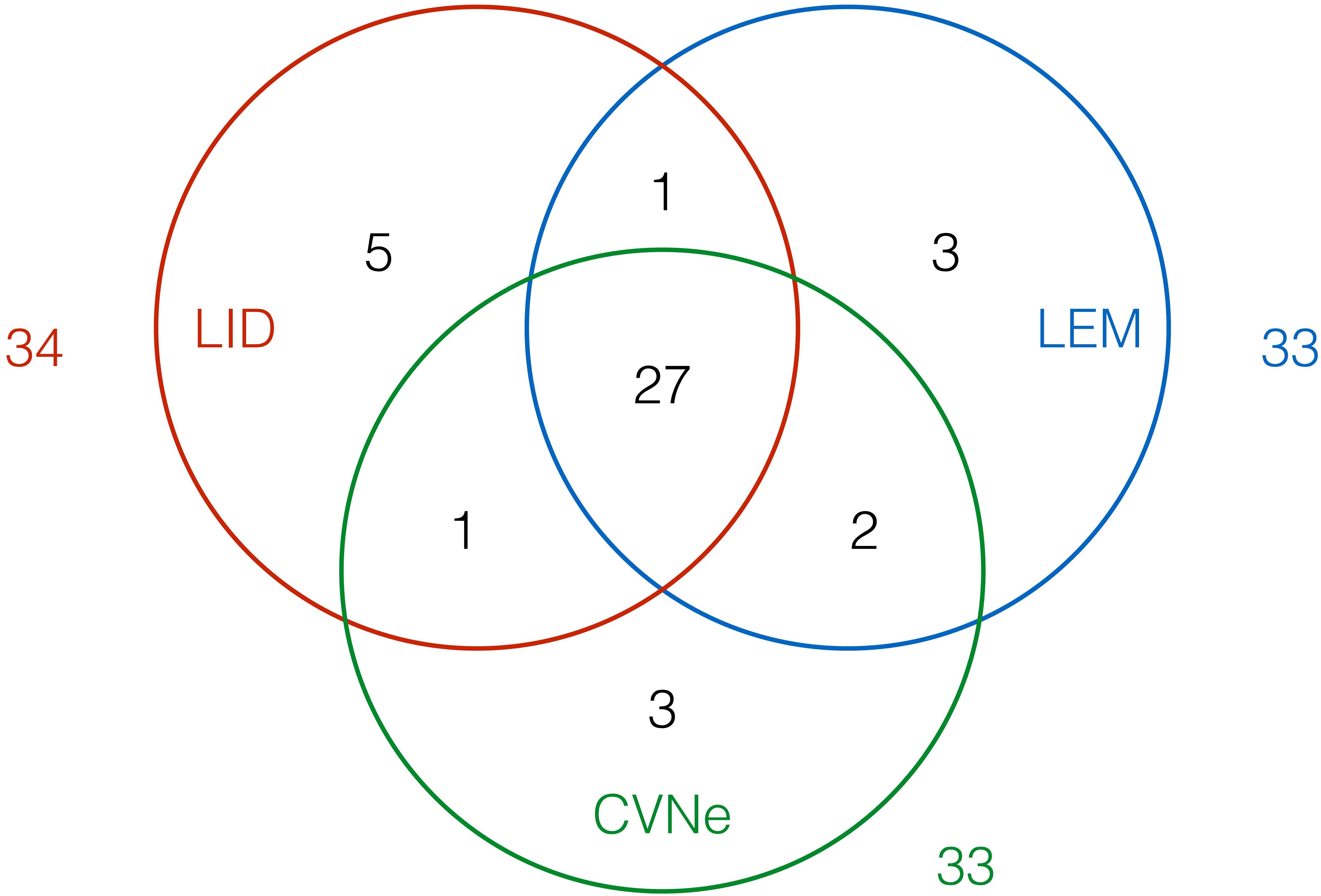
ν_e Event Count



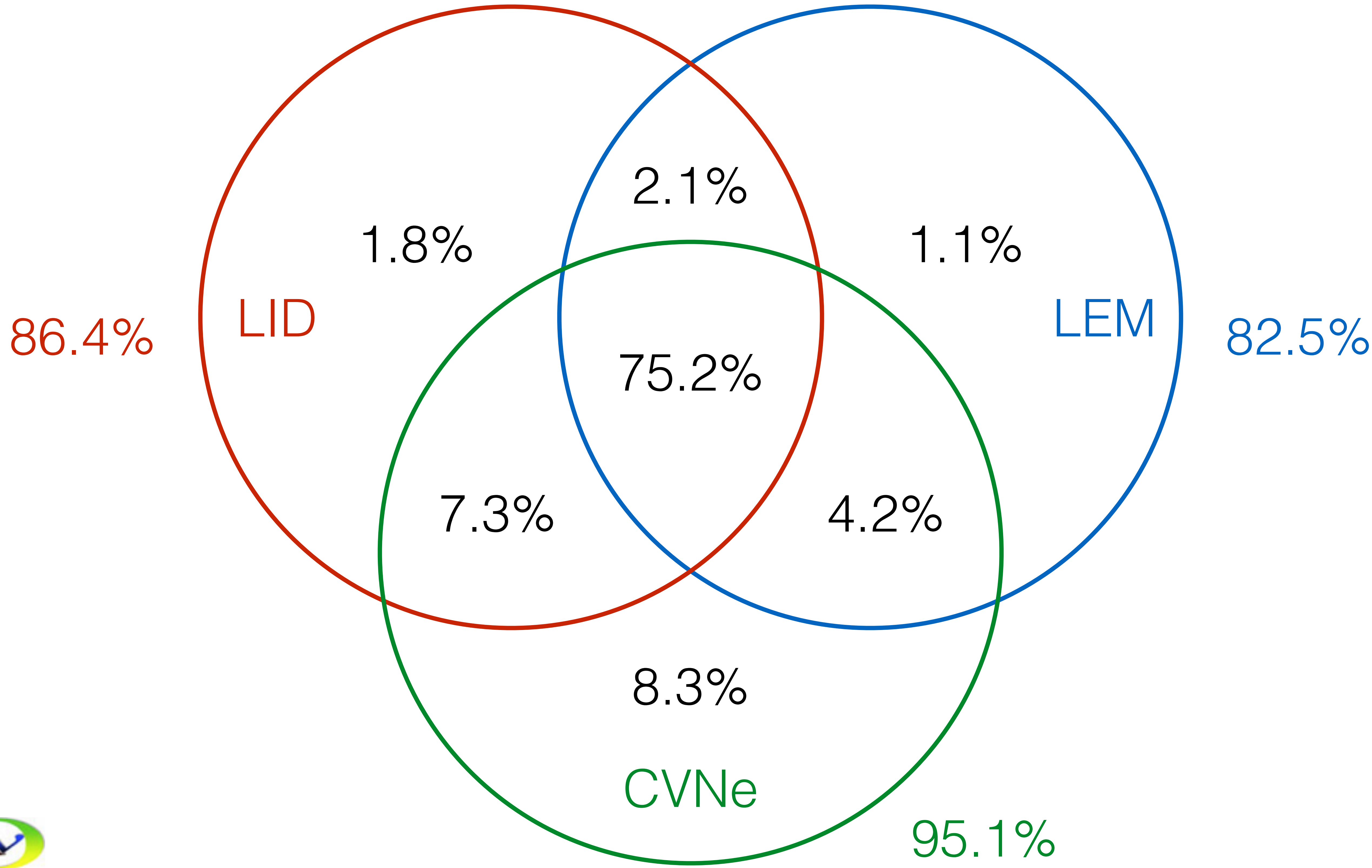


CVN sees 33 events on an expected background of 8.2.
Previous result PIDs: LID(LEM) sees 34(33) events on bkg. of 12.2(10.3)

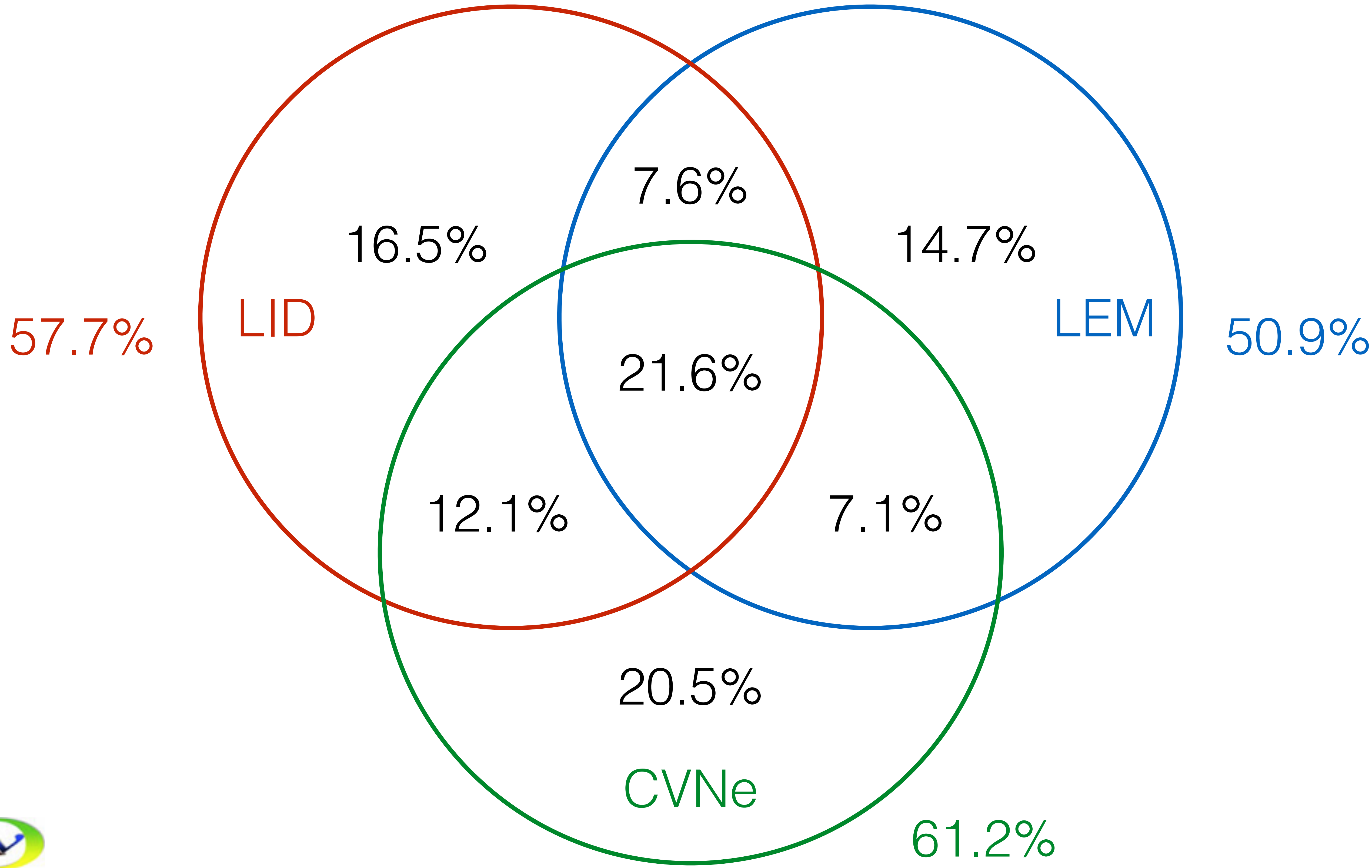
ν_e Data Overlap, S/Sqrt(S+B) Cuts



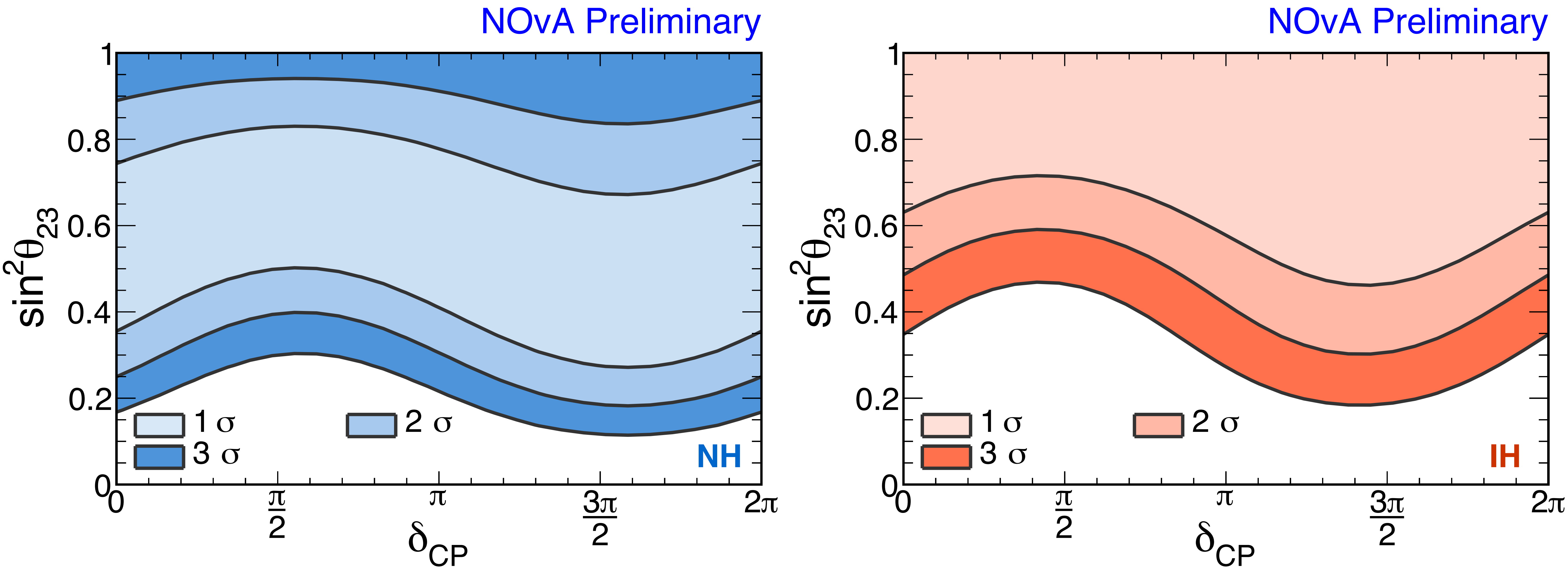
ν_e Expected Signal Overlap, S/Sqrt(S+B) Cuts, Venn Diagram



ν_e Expected Beam Background Overlap, S/Sqrt(B) Cuts, Venn Diagram

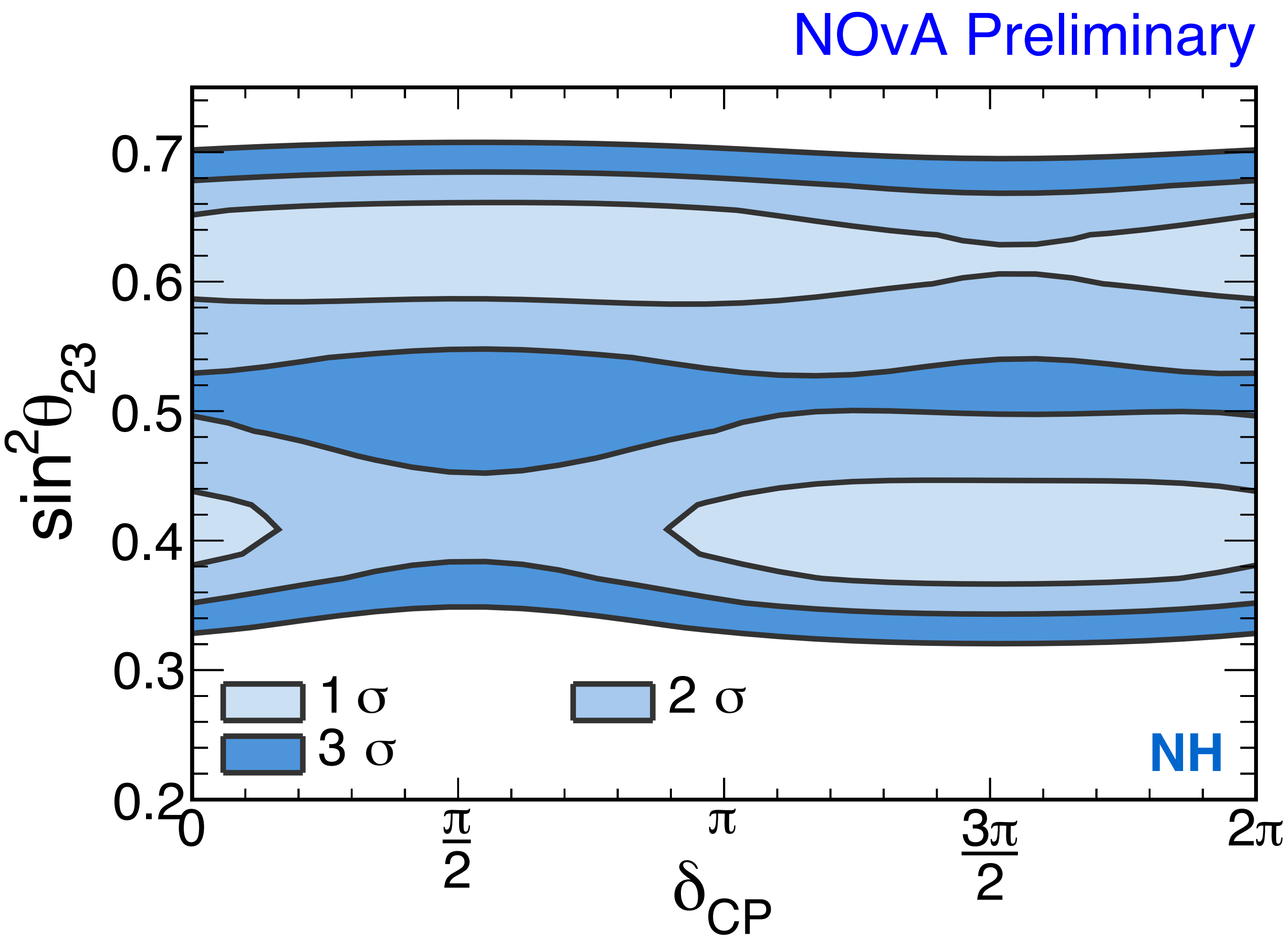


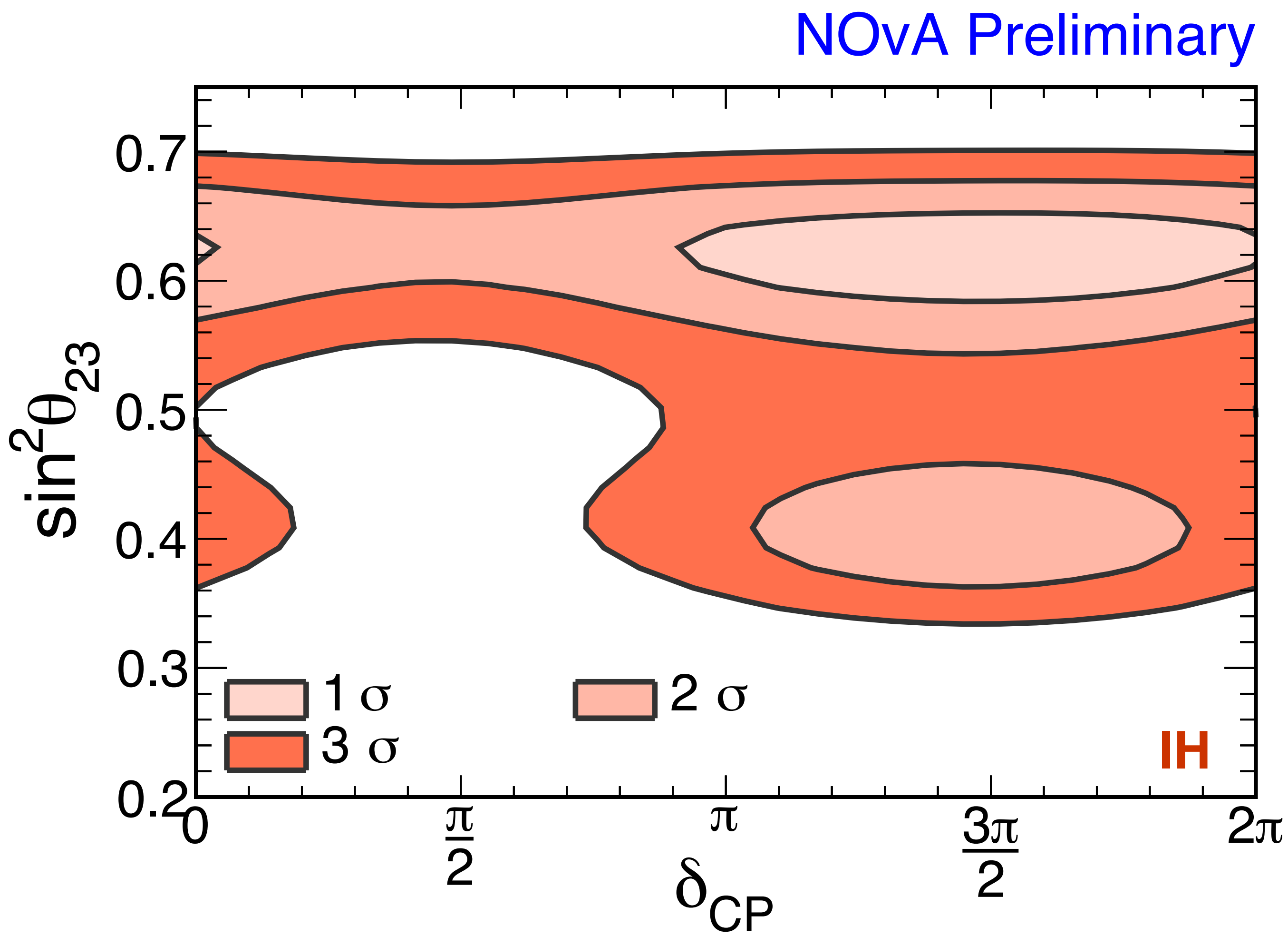
ν_e Appearance Best Fit



Using the reactor constraint on $\sin^2 2\theta_{13} = 0.085 \pm 0.005$

ν_e Appearance Best Fit



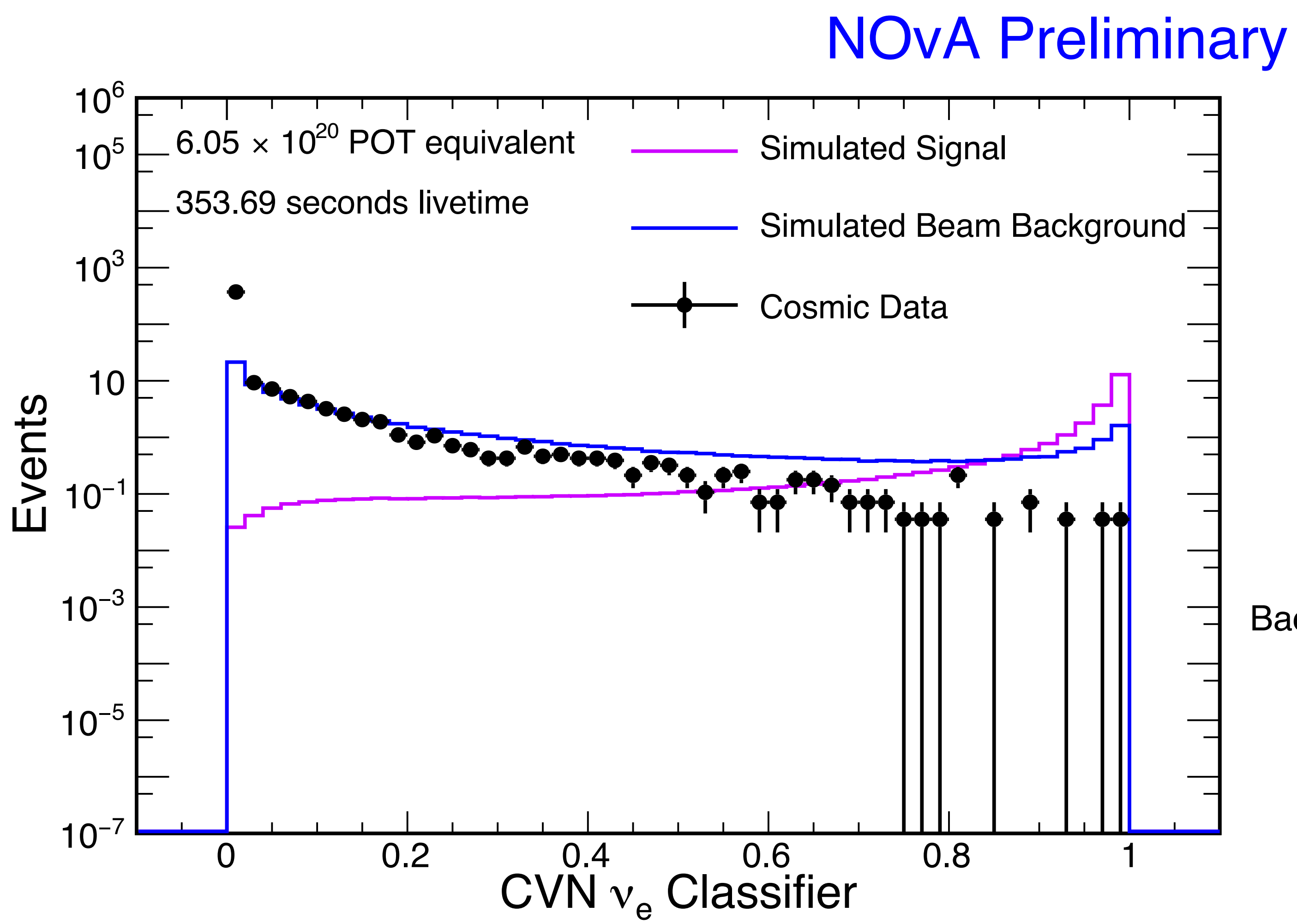


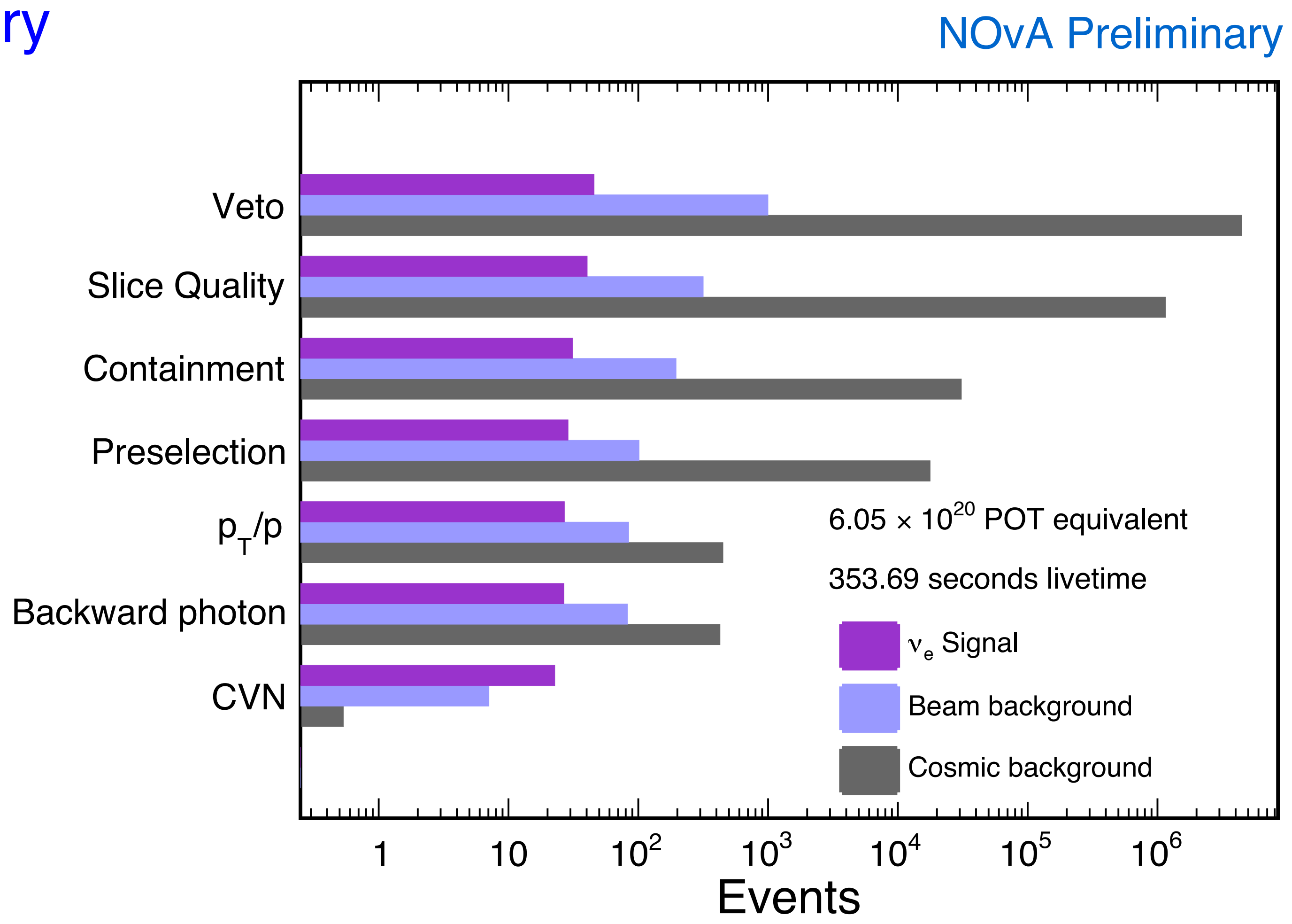
Using the reactor constraint on $\sin^2 2\theta_{13} = 0.085 \pm 0.005$
$\Delta m^2_{32}/\theta_{23}$ from NOvA disappearance result (not a full joint fit)

Small NH preference at $\Delta\chi^2 = 0.46$

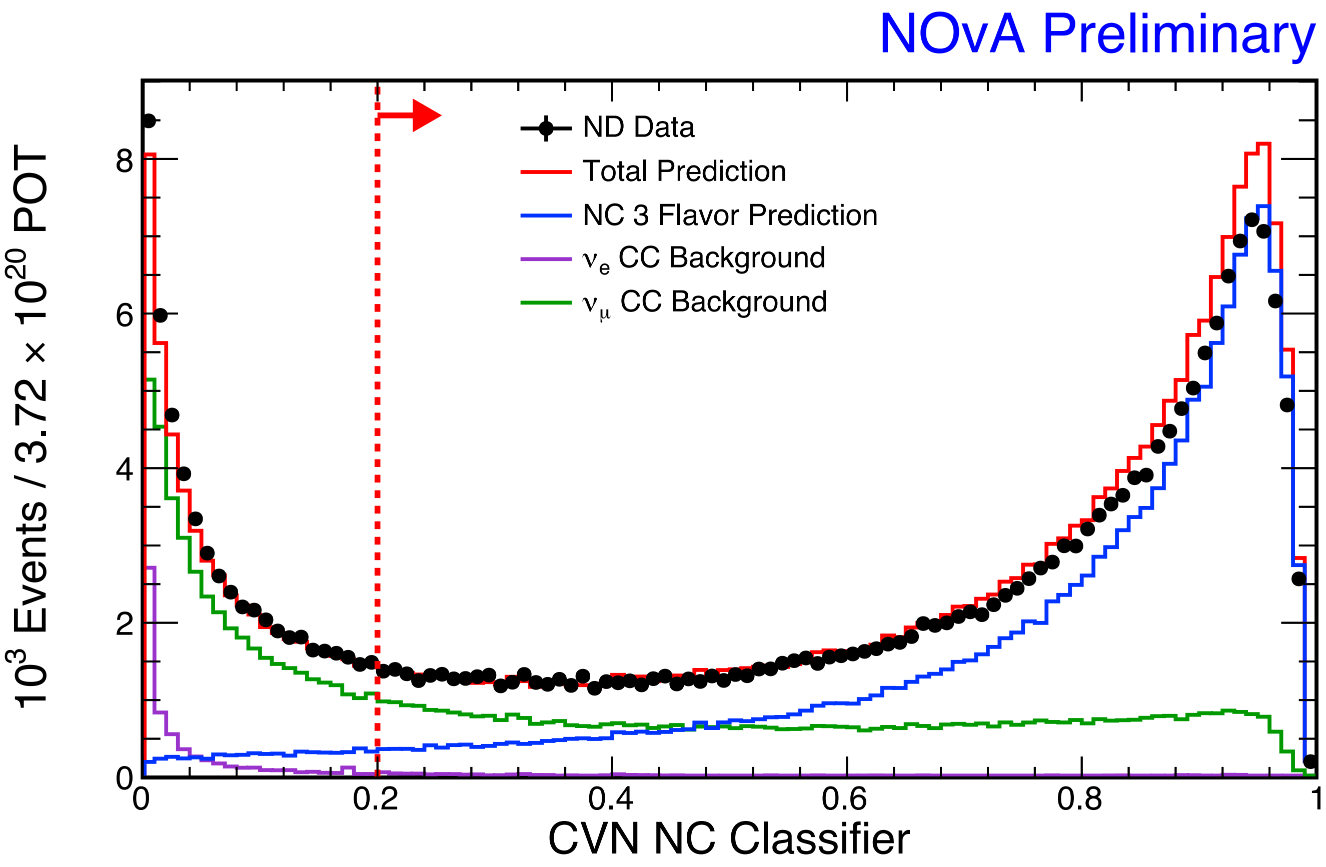
IH $\delta_{CP} = \pi/2$ excluded at 3σ

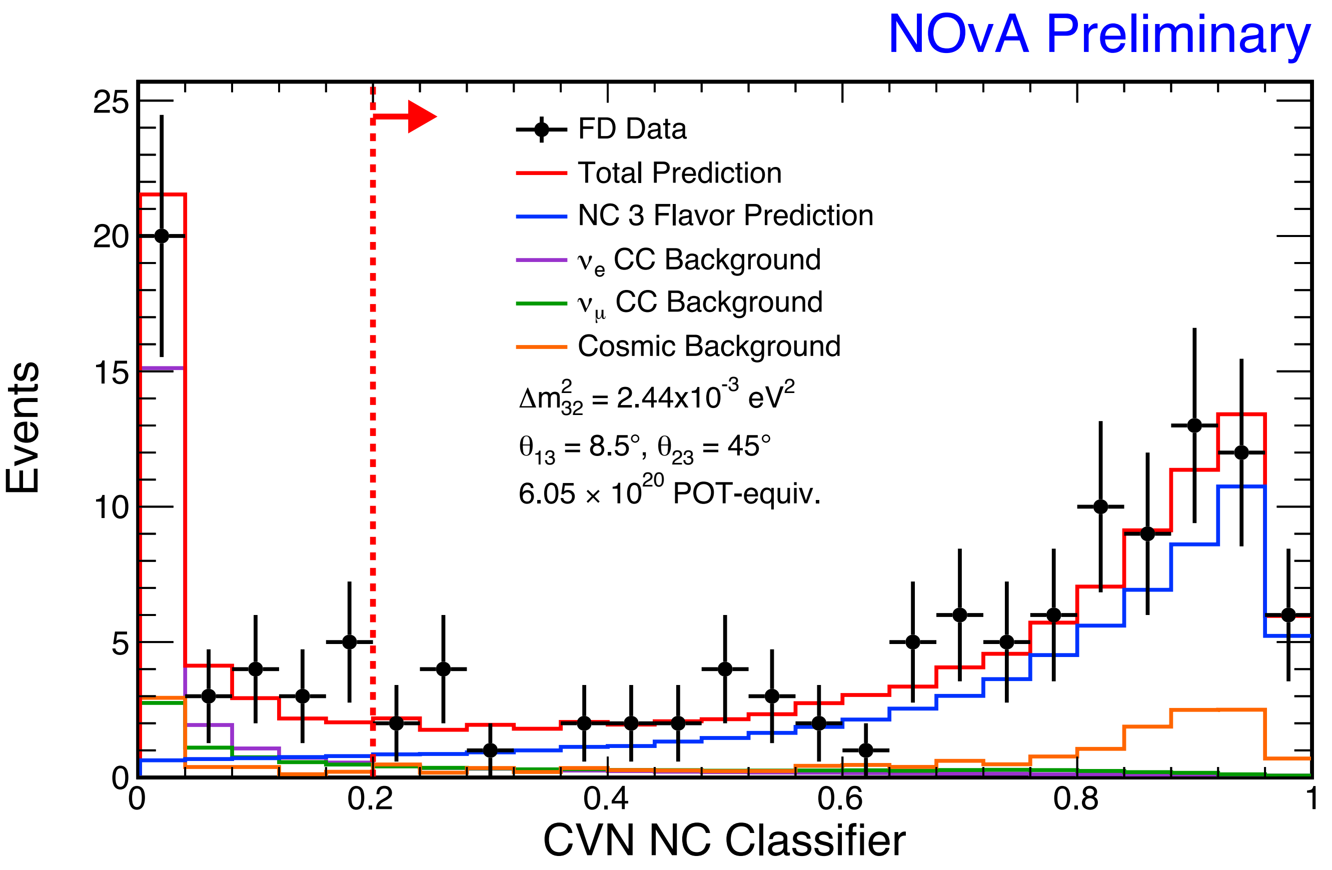
ν_e Cosmic Background



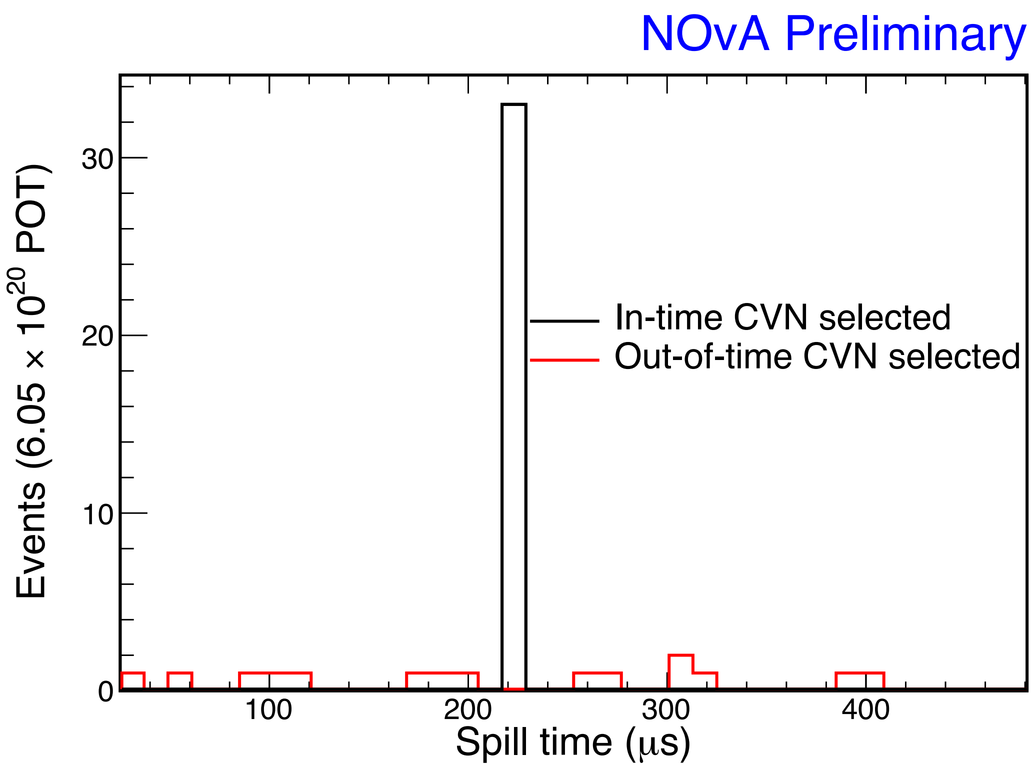


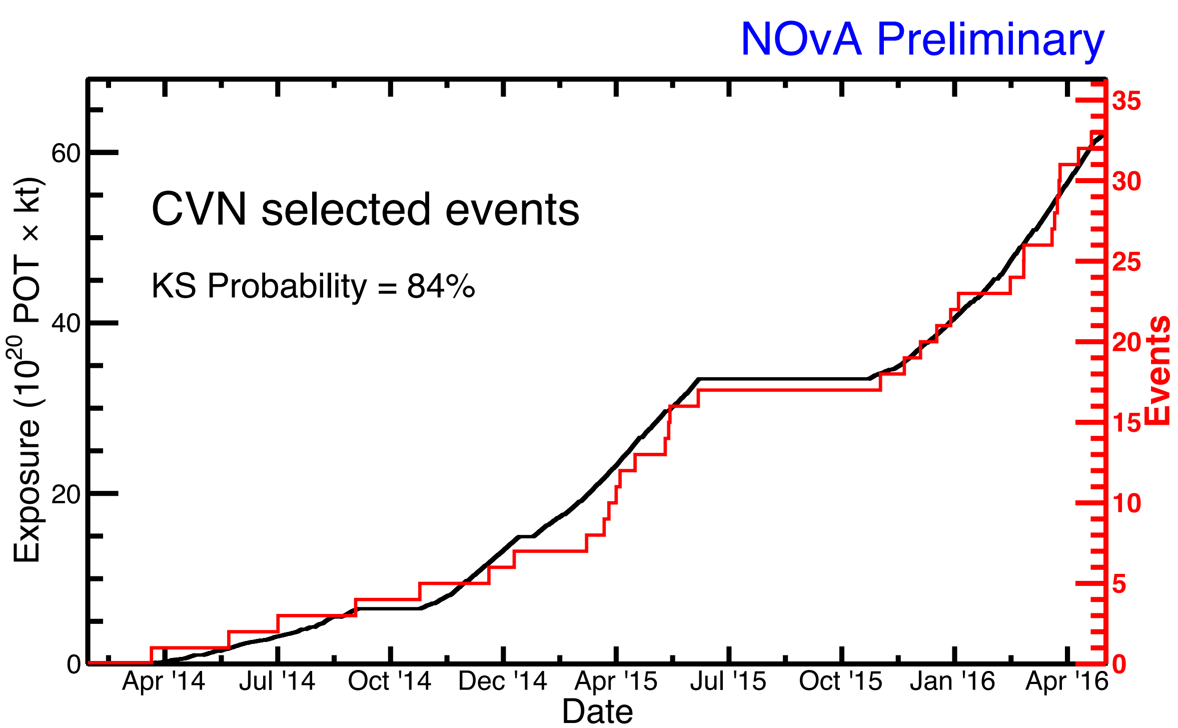
ν_s Near and Far Detector CVN for NC

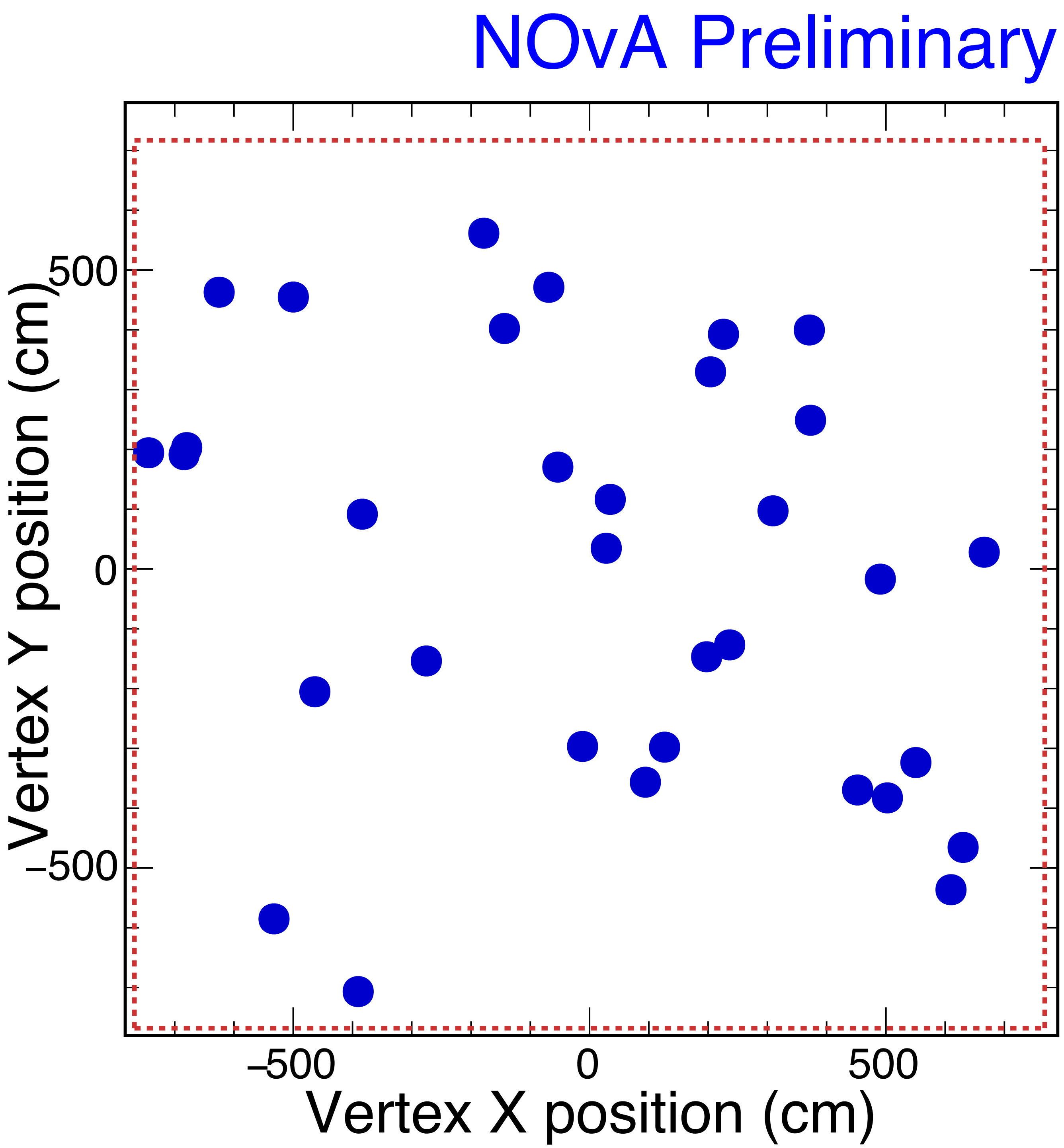




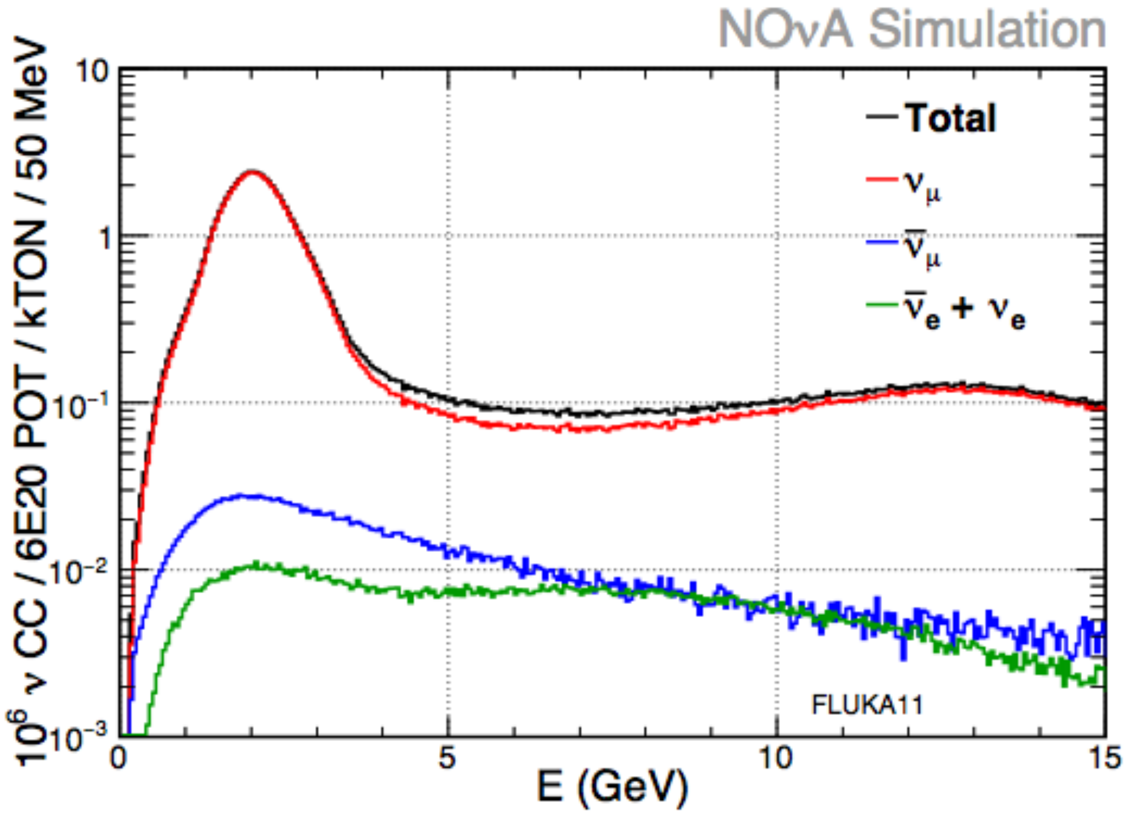
ν_e candidates, when & where?

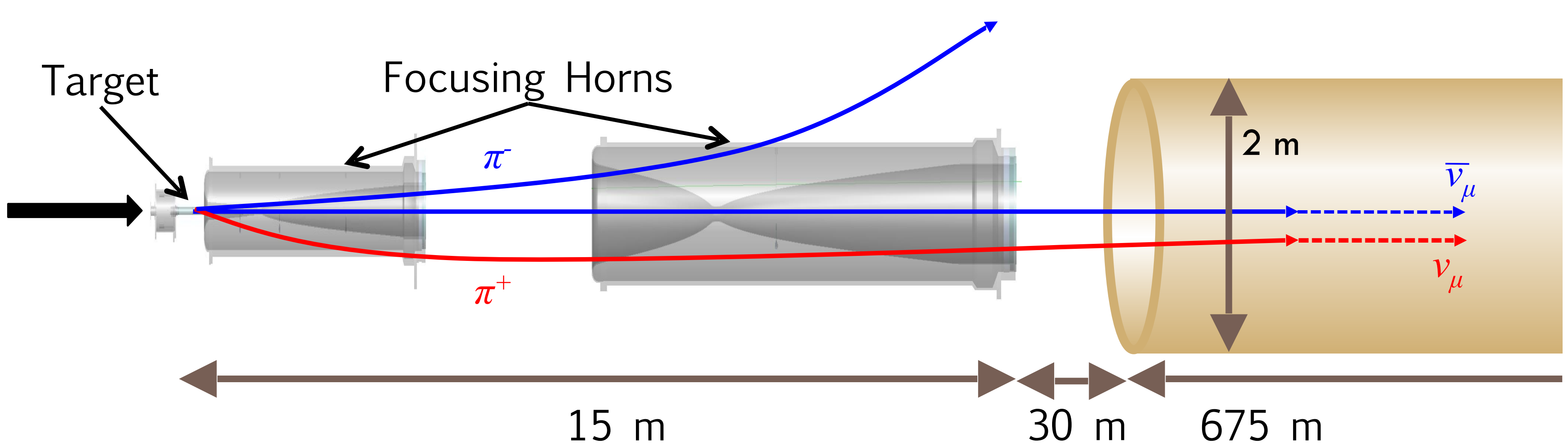




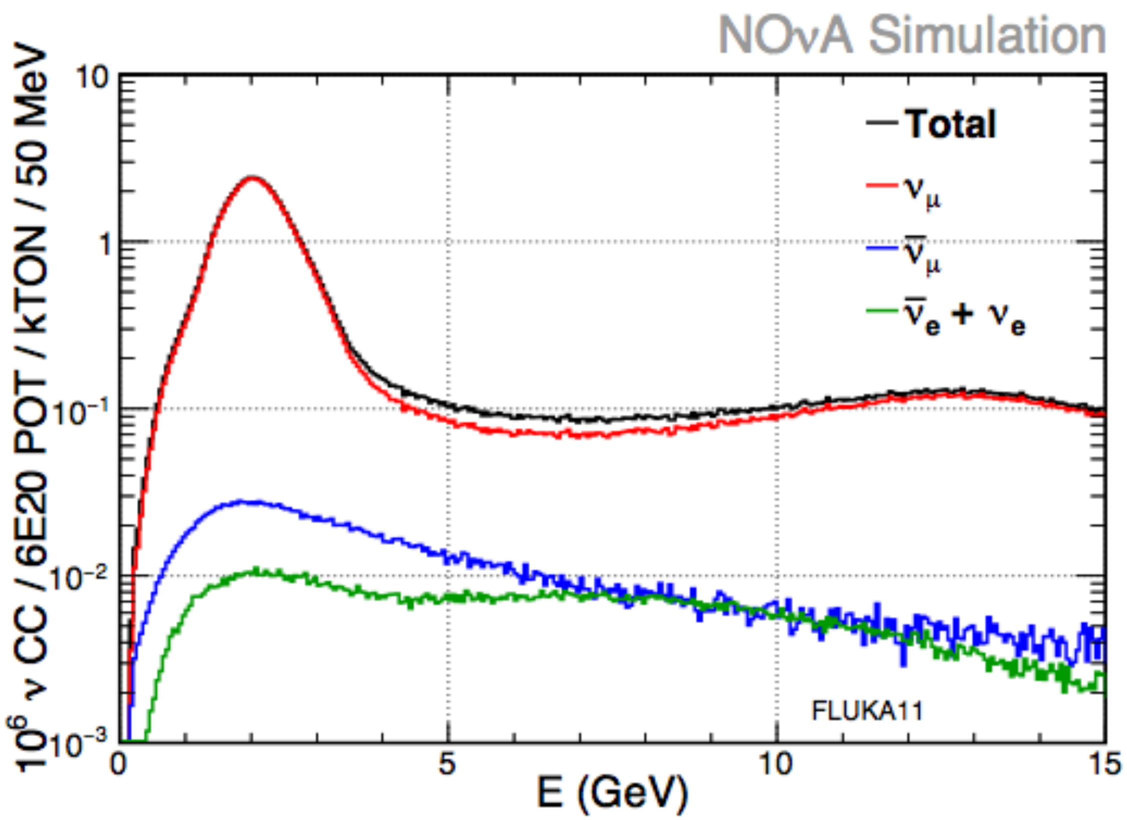


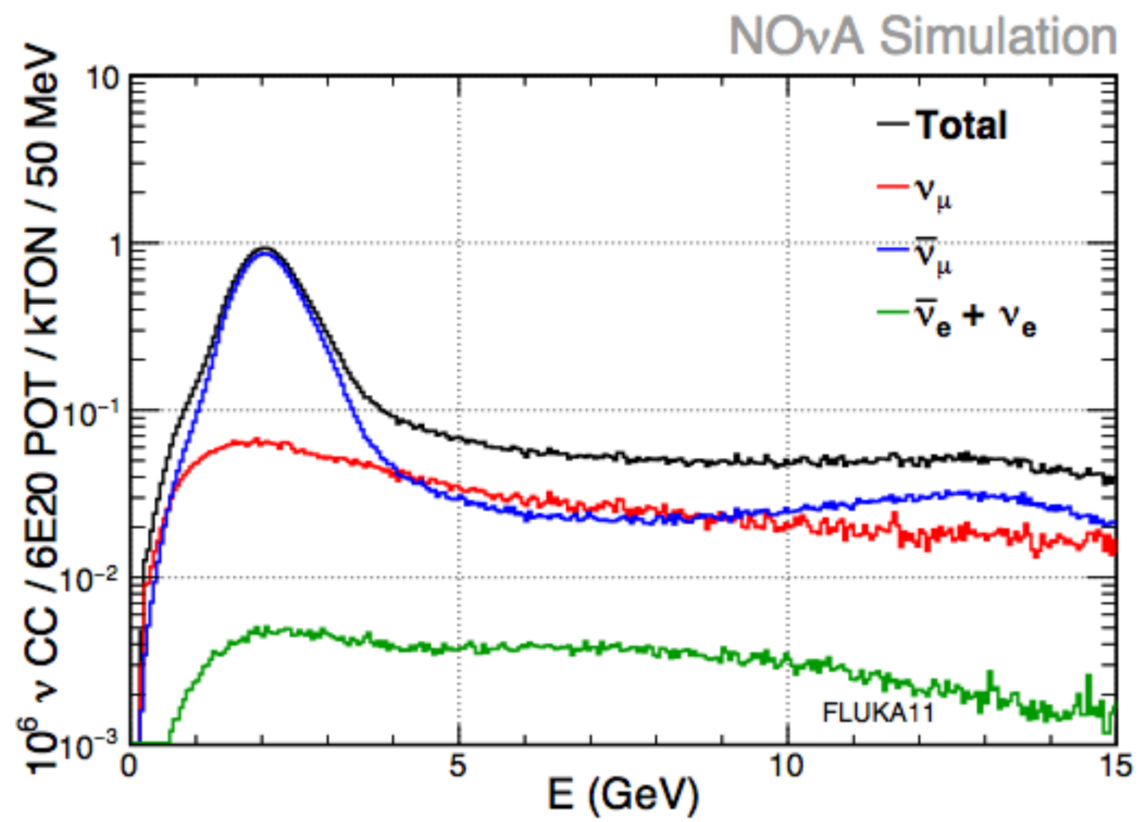
Neutrino Mode

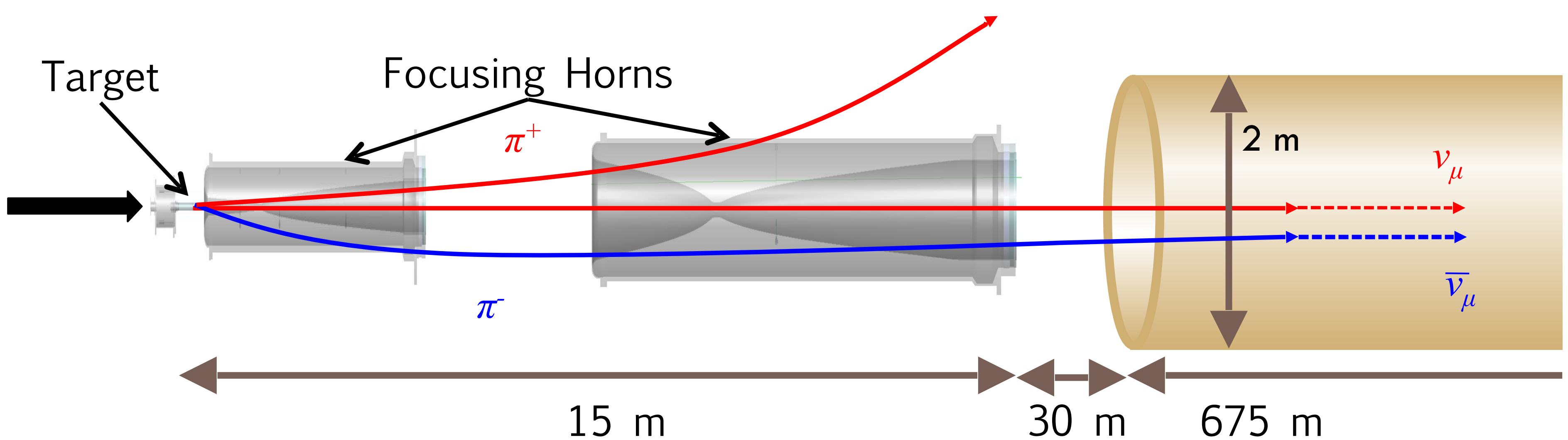




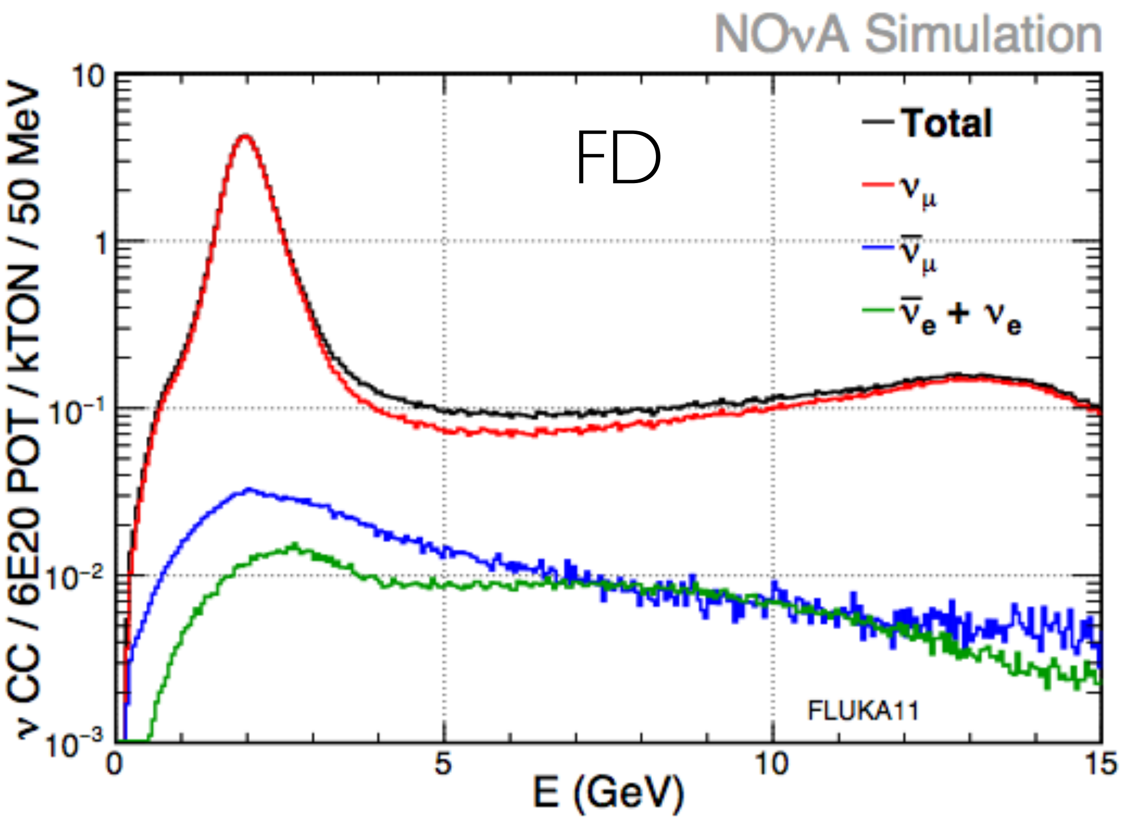
Anti-neutrino Mode

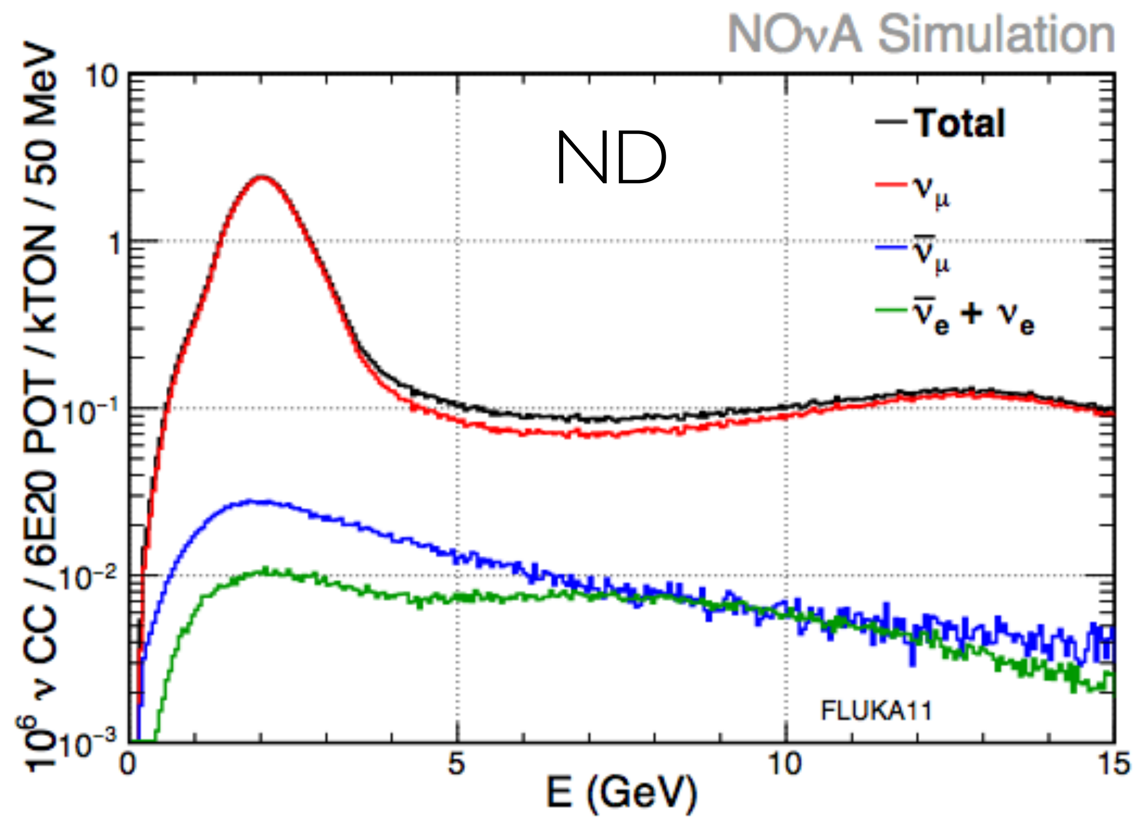






Neutrino Mode Flux Composition





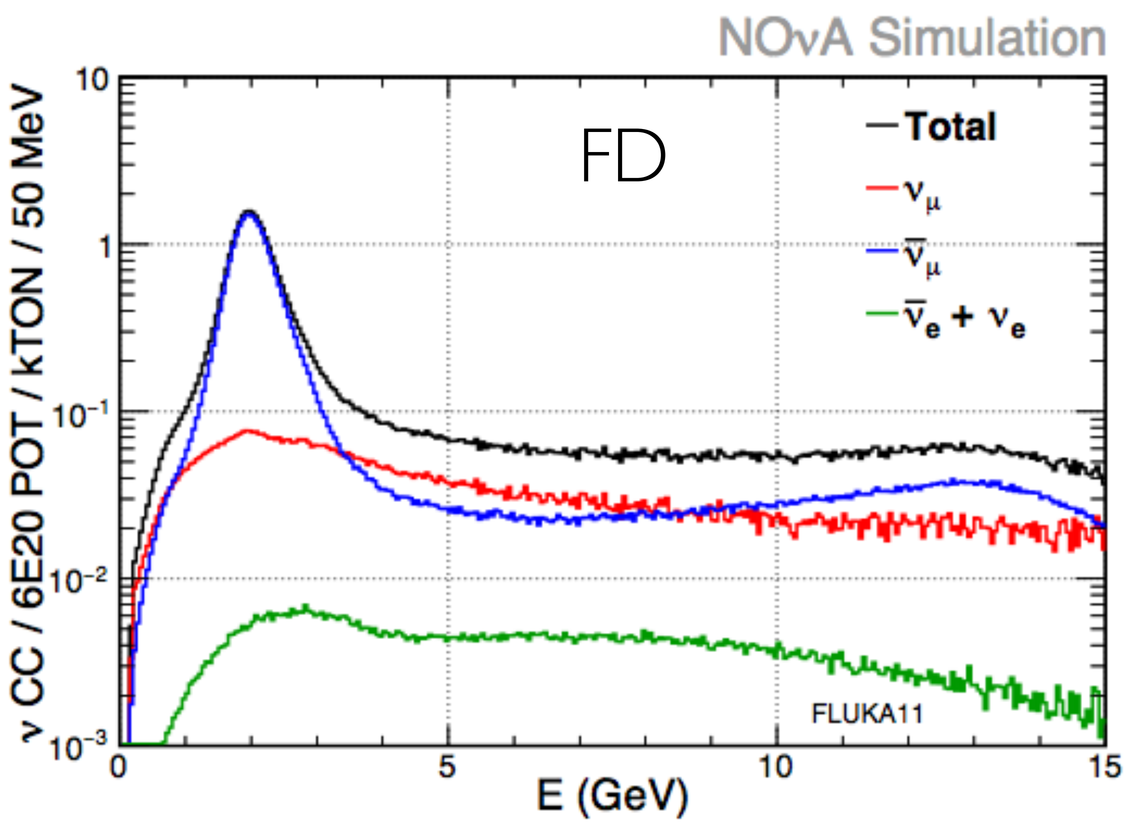
	[1,3]GeV	[0,120]Gev
Total	68.0	109.1
ν_μ	66.5	102.6
$\bar{\nu}_\mu$	1.1	4.1
ν_e —	0.4	2.4

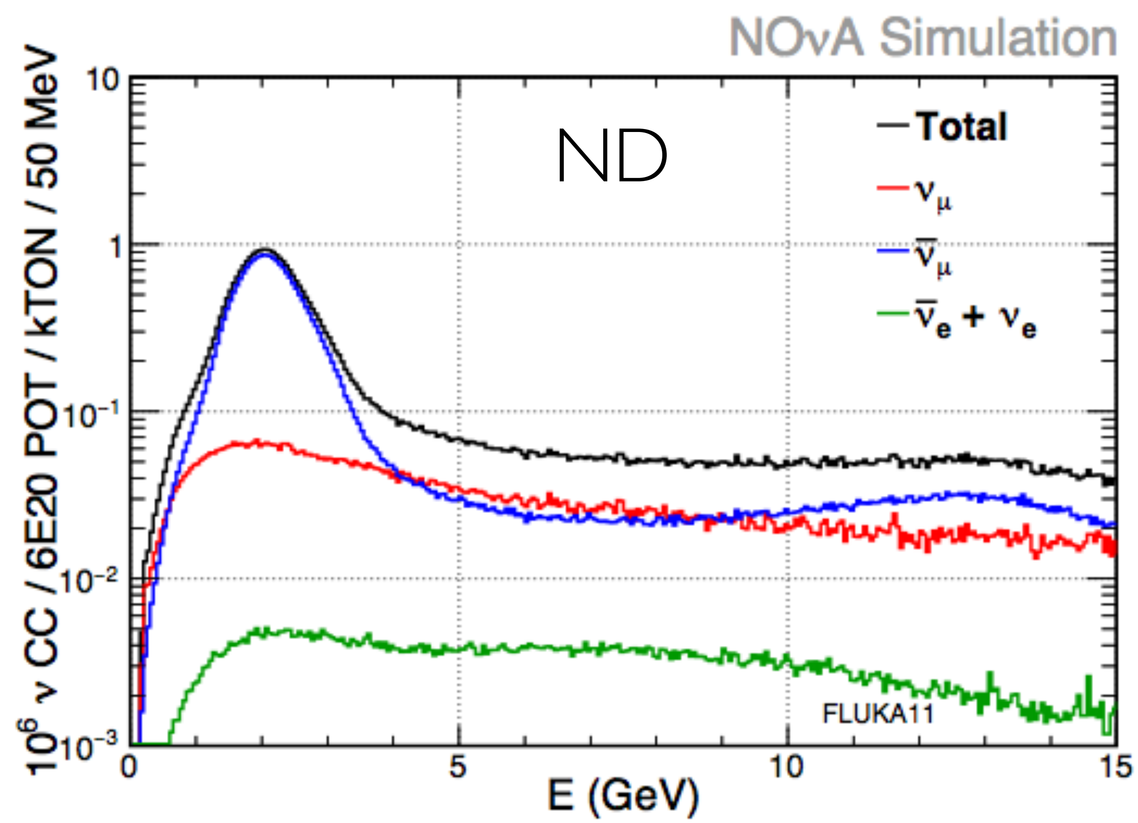
x10	[1,3]GeV	[0,120]Gev
Total	58.7	101.3
ν_μ	57.3	95.4
$\bar{\nu}_\mu$	1.0	3.8
ν_e —	0.4	2.1

[1,3]GeV: $\bar{\nu}_\mu / \nu_\mu$ = 1.7%
[1,3]GeV: $(\nu_e + \bar{\nu}_e)/\nu_\mu$ = 0.6%

[1,3]GeV: $\bar{\nu}_\mu / \nu_\mu$ = 1.7%
[1,3]GeV: $(\nu_e + \bar{\nu}_e)/\nu_\mu$ = 0.7%

Antineutrino Mode Flux Composition





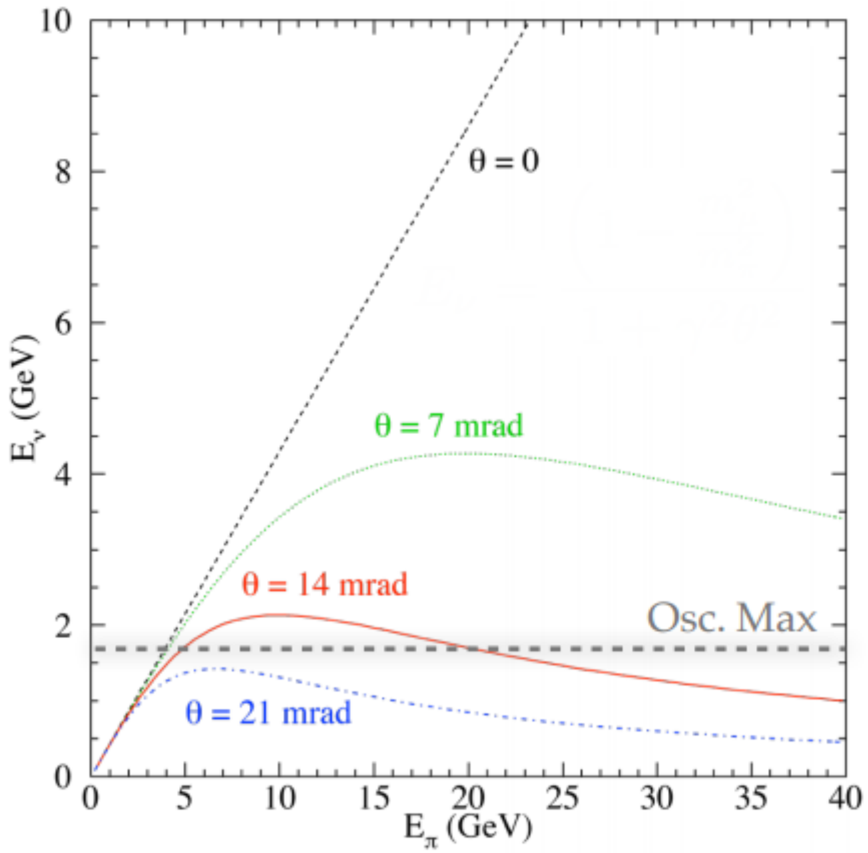
	[1,3]GeV	[0,120]Gev
Total	26.7	48.8
ν_μ	2.7	14.1
$\bar{\nu}_\mu$	23.8	33.4
ν_e	0.2	1.3

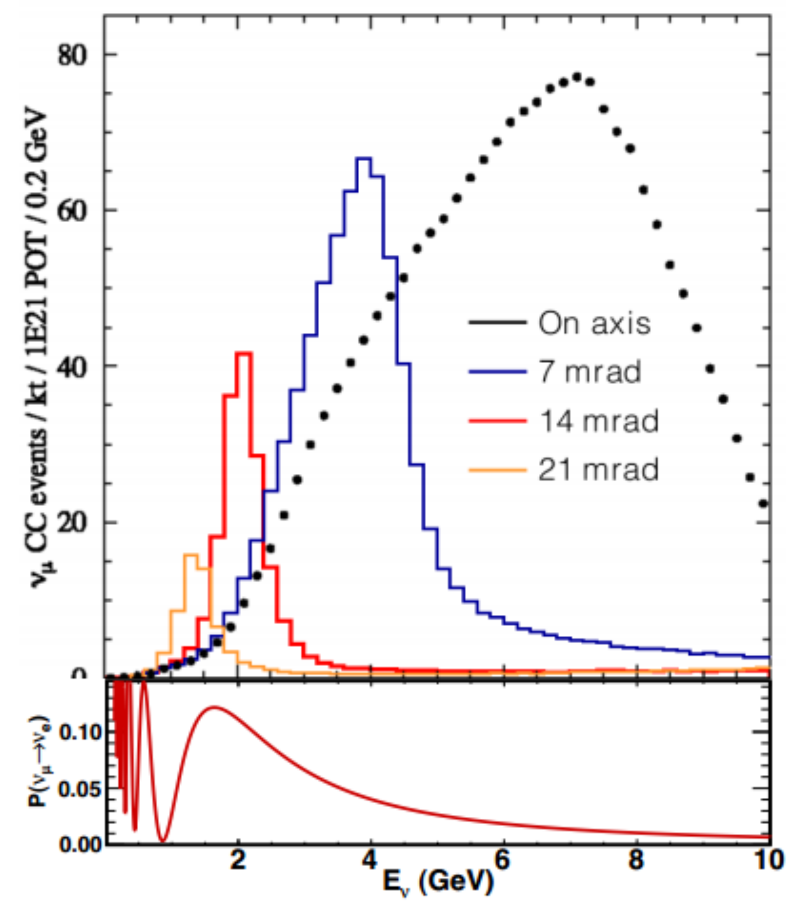
x10	[1,3]GeV	[0,120]Gev
Total	23.1	44.8
ν_μ	2.4	12.9
$\bar{\nu}_\mu$	20.5	30.8
ν_e	0.2	1.1

[1,3]GeV: $\nu_\mu/\bar{\nu}_\mu$ = 11%
[1,3]GeV: $(\nu_e+\bar{\nu}_e)/\bar{\nu}_\mu$ = 0.8%

[1,3]GeV: $\nu_\mu/\bar{\nu}_\mu$ = 12%
[1,3]GeV: $(\nu_e+\bar{\nu}_e)/\bar{\nu}_\mu$ = 1%

Off Axis





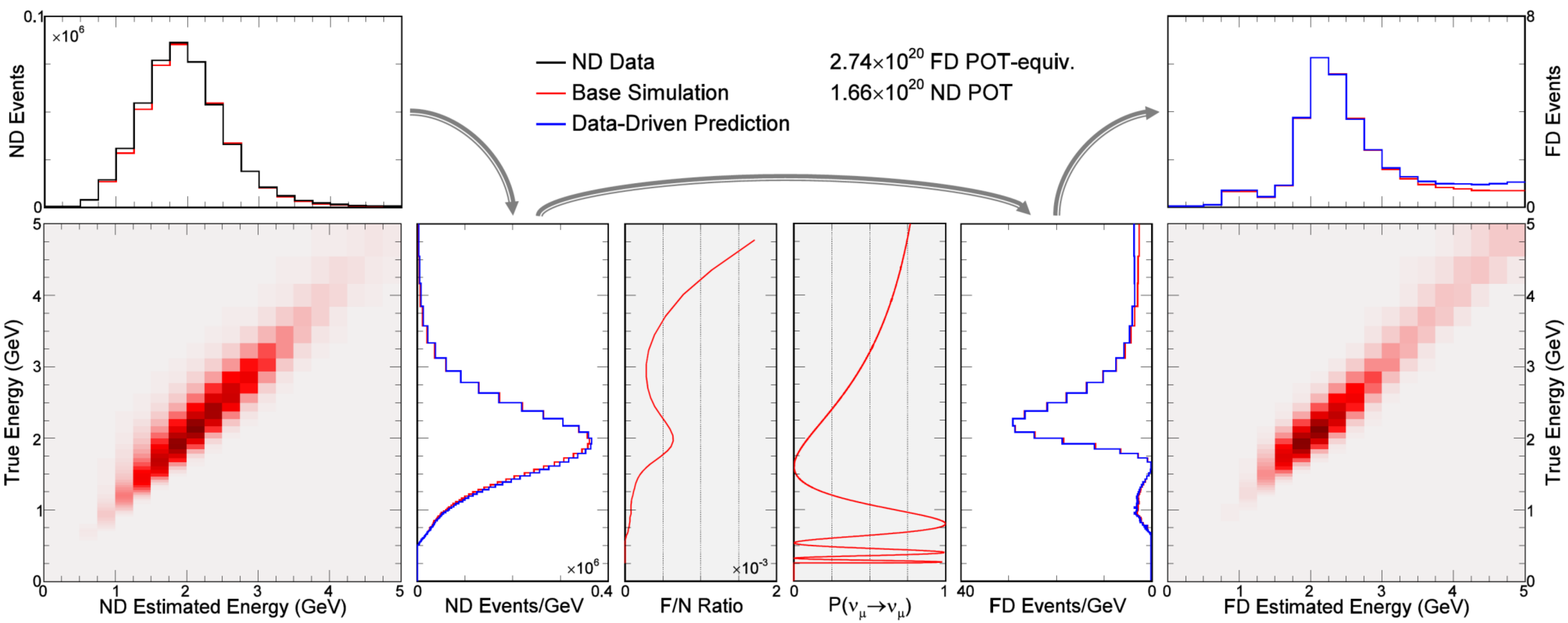
$$F = \left(\frac{2\gamma}{1+\gamma\theta^2}\right)^2 \frac{A}{4\pi z^2}$$

$$E_\nu = \frac{\left(1 - \frac{m_\mu^2}{m_{\pi,K}^2}\right) E_{\pi,K}}{1+\gamma^2\theta^2}$$

Allows us to select lower energy neutrinos, optimized for our baseline. It also means we have a narrowband flux such that we see far fewer neutral currents in the electron neutrino appearance region.

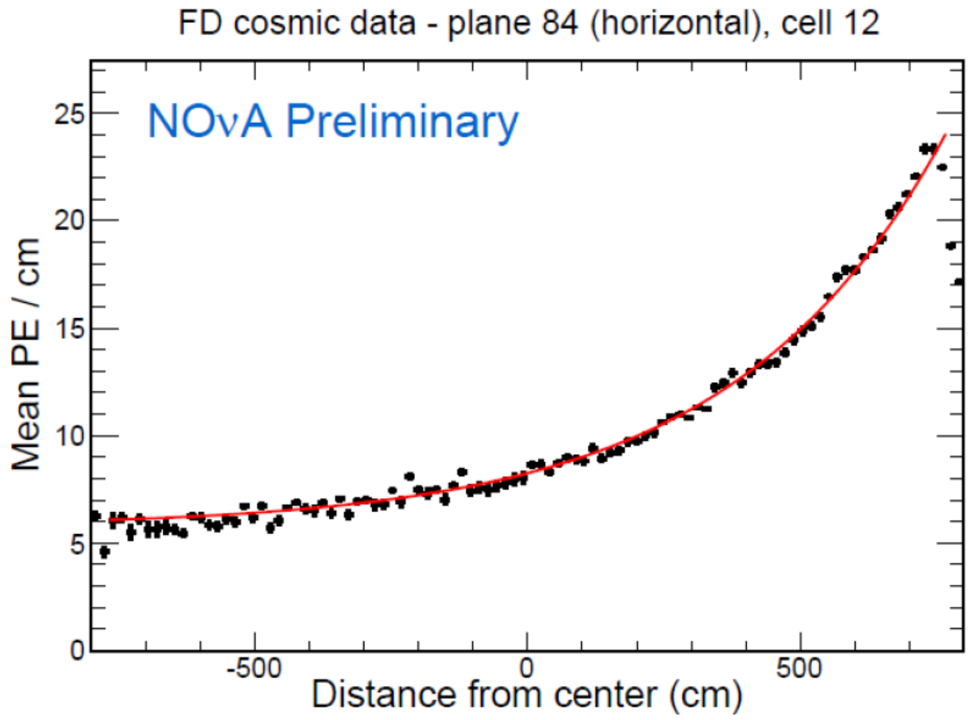
Far Detector Prediction

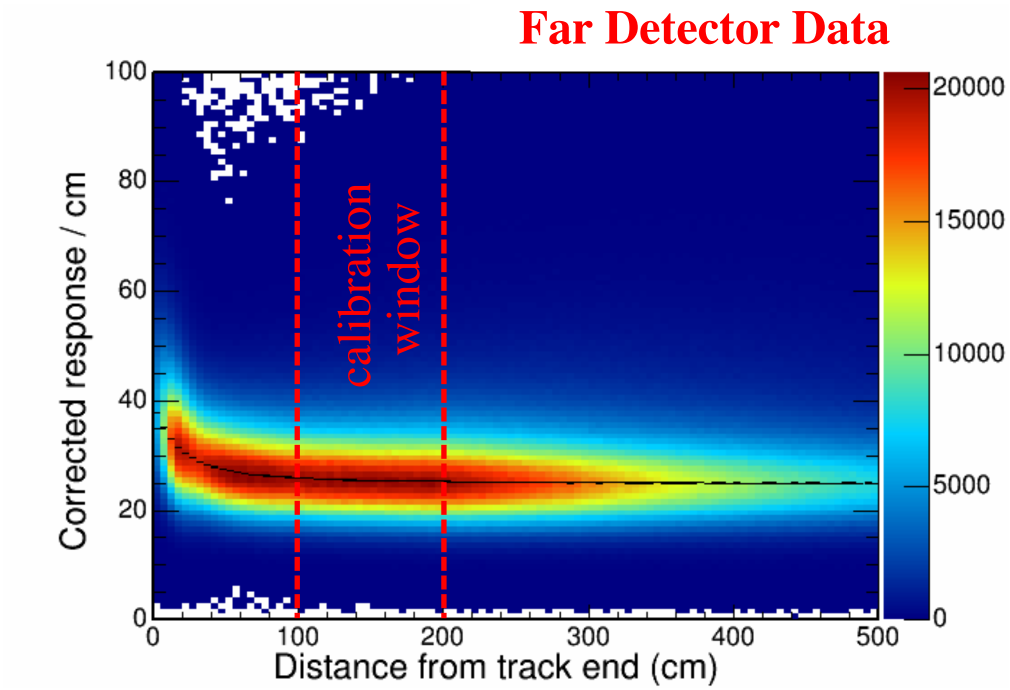
1. Estimate the true energy distribution of selected ND events
2. Multiply by expected Far/Near event ratio and $\nu_\mu \rightarrow \nu_\mu$ oscillation probability as a function of true energy
3. Convert FD true energy distribution into predicted FD reco energy distribution with systematic assessed by varying all MC based steps

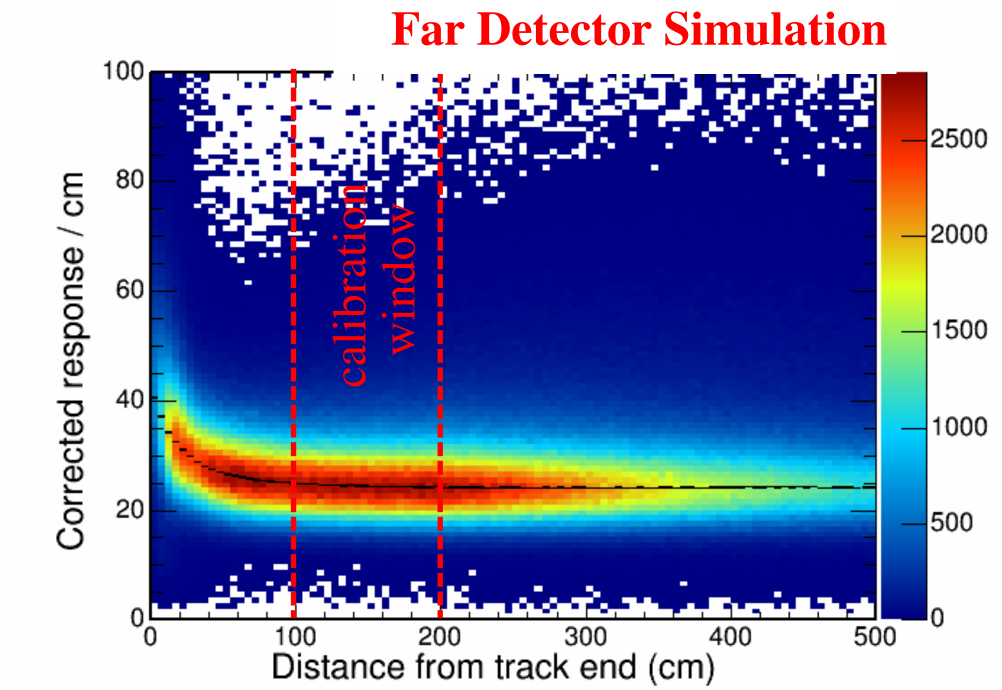


Calibration

- Largest effect that needs correction is attenuation in the WLS fibre
- Stopping cosmic muons provide a standard candle for setting absolute energy scale





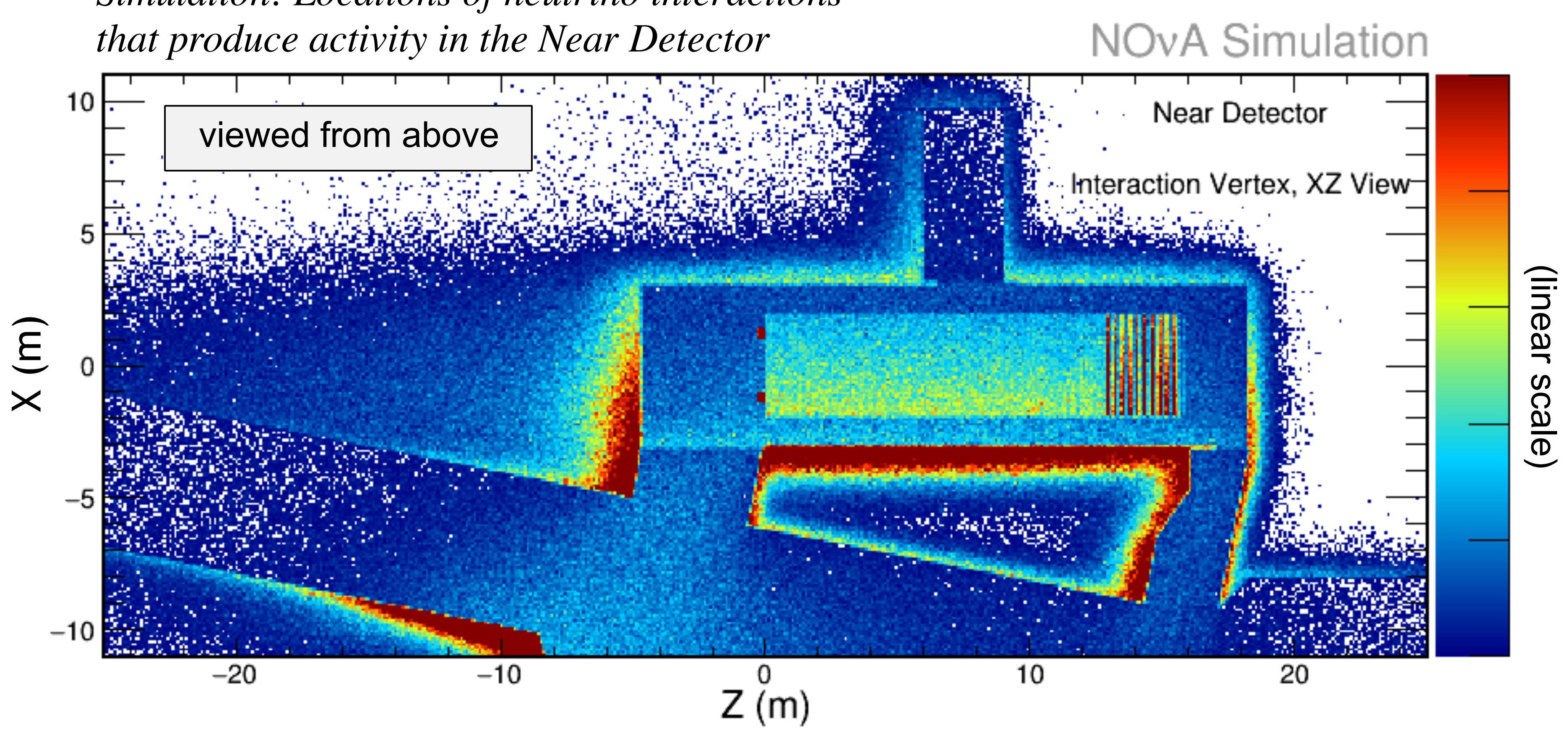


Simulation

Highly detailed end-to-end simulation chains:

- Beam hadron production, propagatio, neutrino flux: **FLUKA/FLUGG**
- Cosmic ray flux: **CRY**
- Neutrino Interactions and FSI modeling: **GENIE**
- Detector Simulation: **GEANT4**
- Readout electronics and DAQ: **Custom simulation routines**

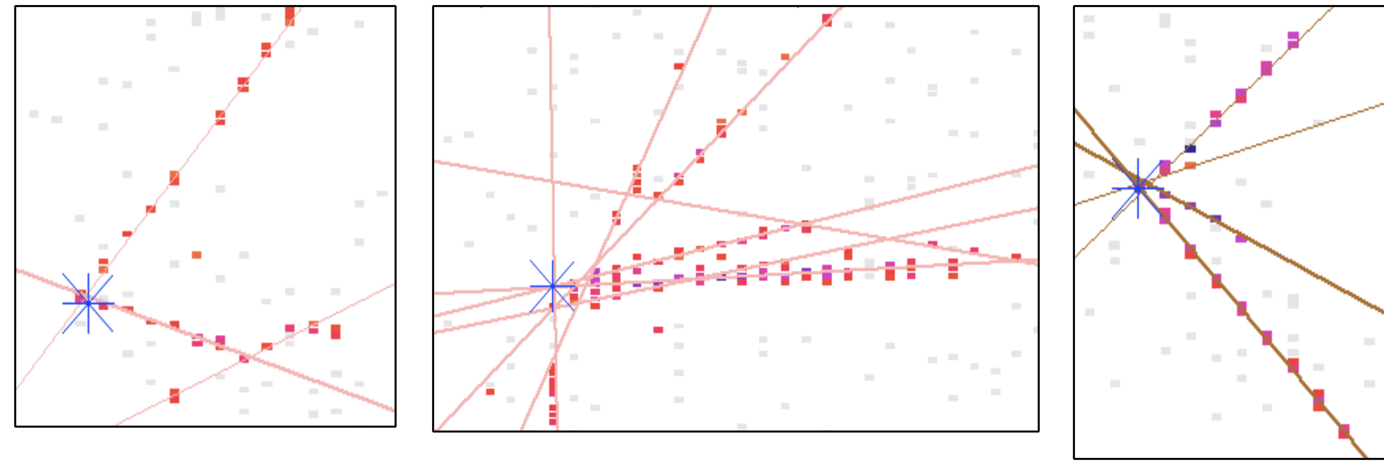
Simulation: Locations of neutrino interactions that produce activity in the Near Detector

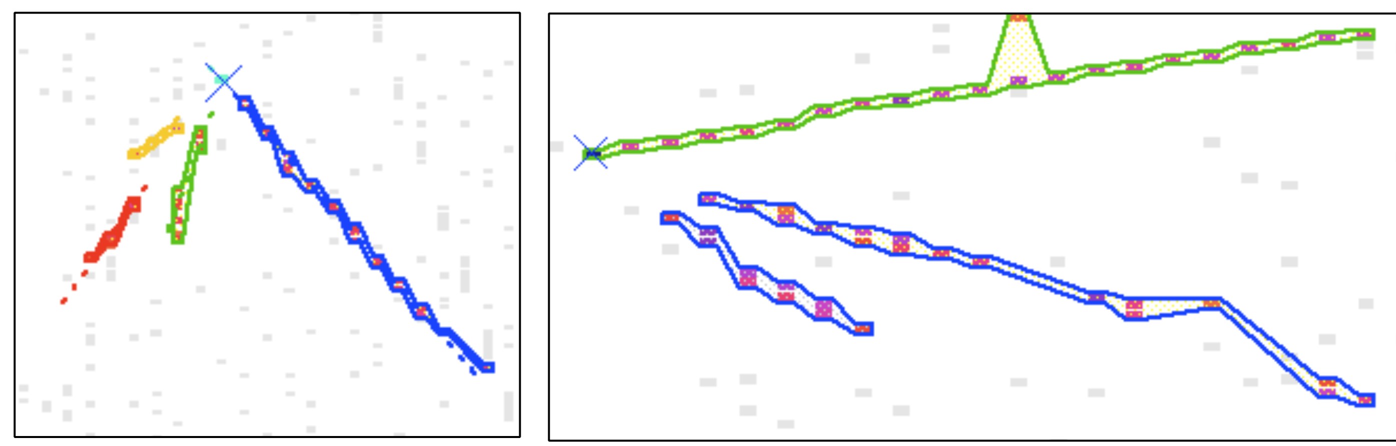


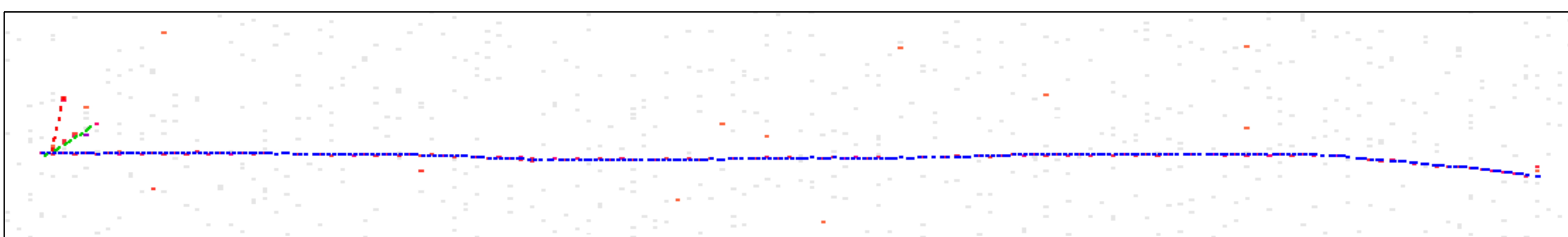
Reconstruction

Three key pieces:

- **Vertexing:** use lines of energy deposition formed with hough transforms to find intersections
- **Clustering:** find clusters in angular space around the vertex and merge views via topology and prong dE/dx
- **Tracking:** Trace particle trajectories using a kalman filter, example below



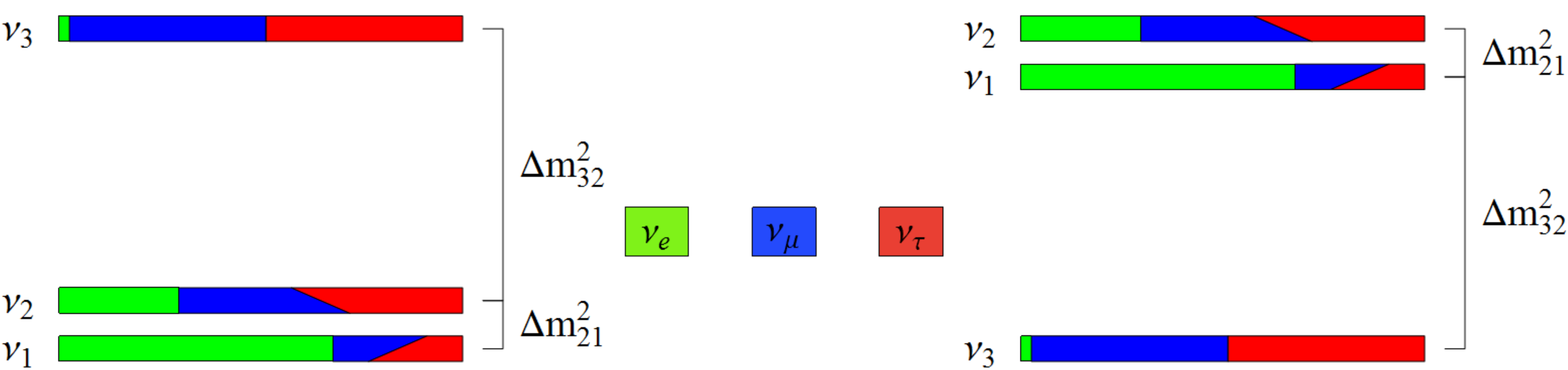




Neutrino Oscillations 101

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = U^* \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} \qquad P(\nu_\alpha \to \nu_\beta) = \left| \sum_j U^*_{\beta j} e^{-i \frac{m_j^2 L}{2E}} U_{\alpha j} \right|^2$$

$$U = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix}$$

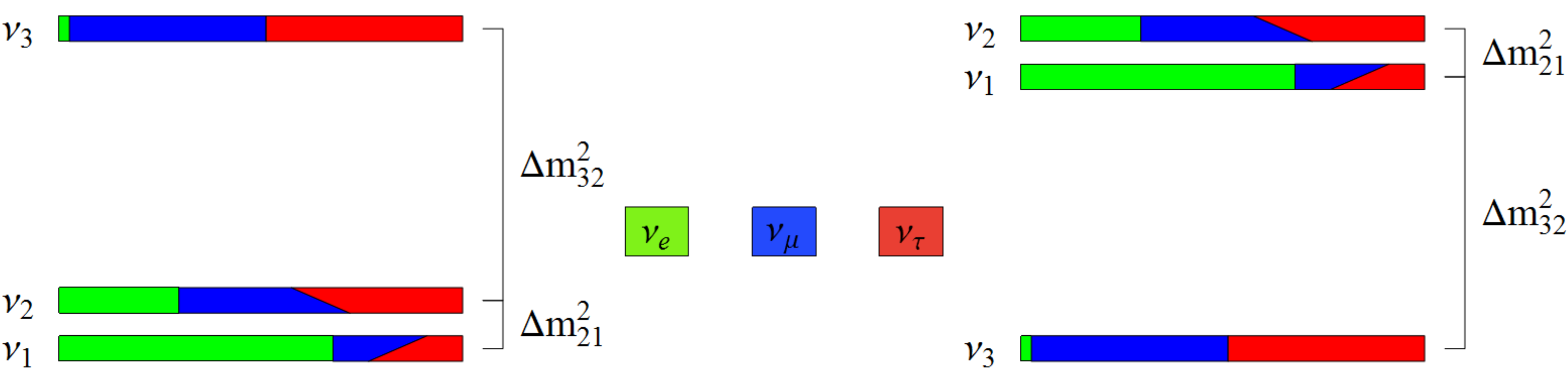


Neutrino Oscillations 101

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = U^* \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix}$$

$$P\left(\nu_\alpha \rightarrow \nu_\beta\right) = \left| \sum_j U^*_{\beta j} e^{-i \frac{m_j^2 L}{2E}} U_{\alpha j} \right|^2$$

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \times \begin{pmatrix} c_{13} & 0 & s_{13} e^{-i\delta} \\ 0 & 1 & 0 \\ -s_{13} e^{i\delta} & 0 & c_{13} \end{pmatrix} \times \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} e^{i\alpha_1/2} & 0 & 0 \\ 0 & e^{i\alpha_2/2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

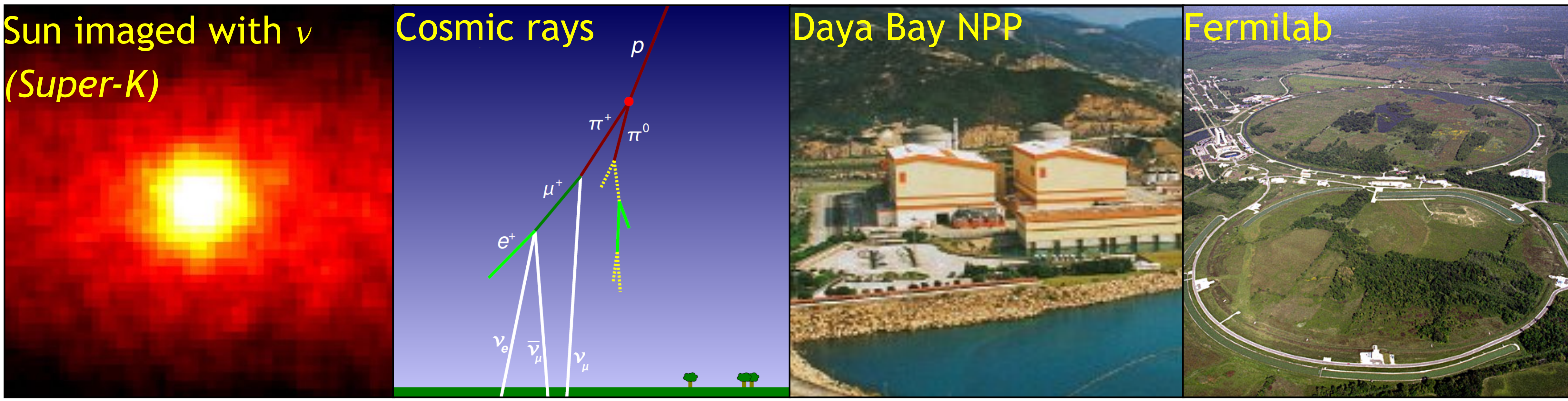


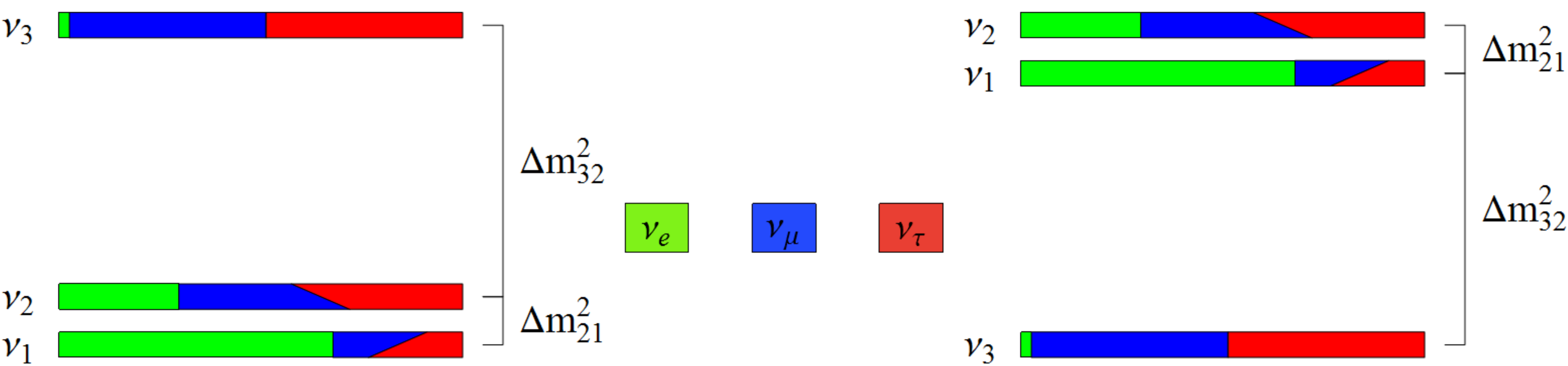
Neutrino Oscillations 101

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = U^* \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix}$$

$$P(\nu_\alpha \to \nu_\beta) = \left| \sum_j U^*_{\beta j} e^{-i \frac{m_j^2 L}{2E}} U_{\alpha j} \right|^2$$

$U =$





Why Study Neutrino Oscillations?

Neutrino oscillations raises as many questions as it answers:

- Why is lepton sector mixing much larger than quark sector mixing?
- What is the hierarchy of neutrino masses and how does this affect searches for a majorana neutrino?
- Is there CP violation in the lepton sector? Could it be large enough to explain observed matter antimatter asymmetry of our universe?

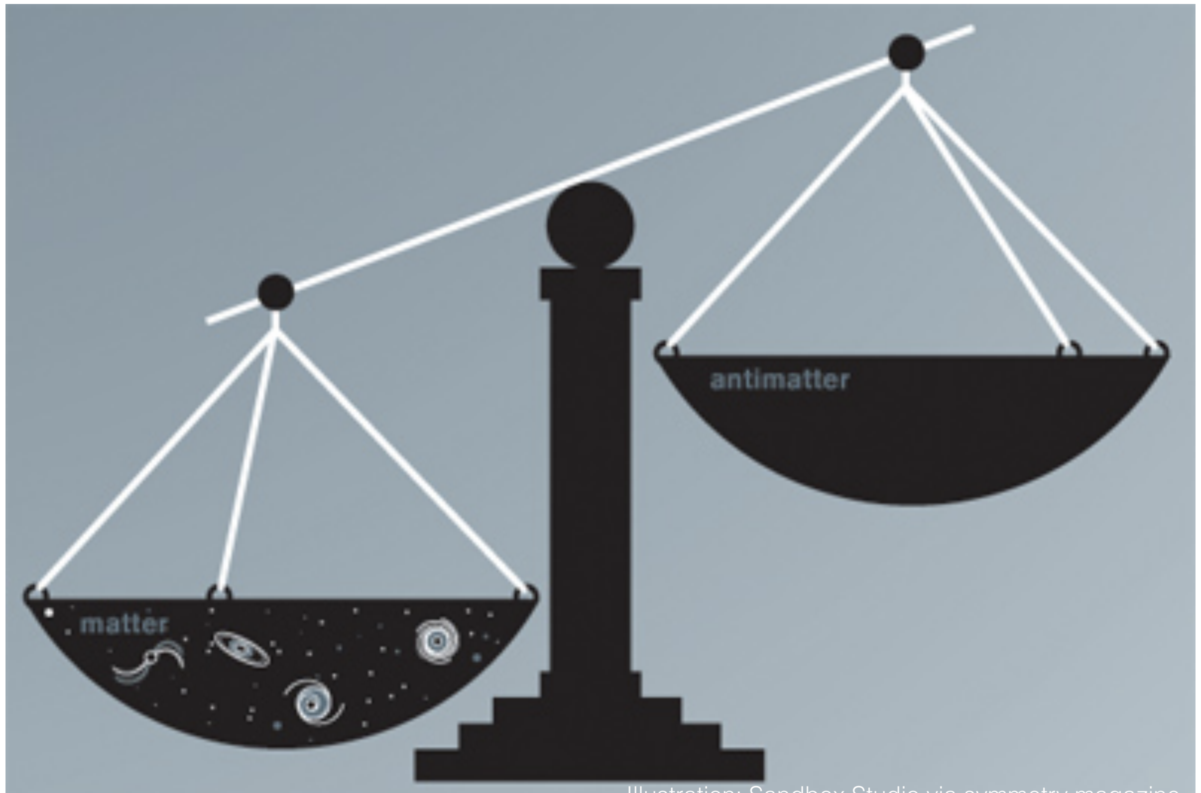


Illustration: Sandbox Studio via symmetry magazine